SIEMENS

SIMATIC

STEP 7
Von S5 nach S7

Umsteigerhandbuch

Ausgabe 03/2006

A5E00706928-01

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:



Gefahr

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

bedeutet, daß eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Marken

SIMATIC® , SIMATIC NET® und SIMATIC HMI® sind eingetragene Marken der SIEMENS AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Siemens AG
Bereich Automation and Drives
Geschäftsgebiet Industrial Automation Systems
Postfach 4848, D- 90327 Nürnberg

Haftungsausschluß

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Wichtige Hinweise

Zweck des Handbuchs

Dieses Handbuch unterstützt Sie beim Umstieg von S5 nach S7.

Die Informationen dieses Handbuchs ermöglichen es Ihnen:

- vorhandene S5-Programme mit einem Konverter in S7-Programme umzusetzen und bei Bedarf manuell nachzubearbeiten.
- bereits konvertierte S7-Funktionen (ehemalige S5-Standard-Funktionsbausteine) in Ihre S7-Programme einzubinden.

Leserkreis

Dieses Handbuch richtet sich an Programmierer, die vorhandene S5-Programme bei S7 einsetzen möchten.

Gültigkeitsbereich des Handbuchs

Dieses Handbuch ist gültig für die Programmiersoftware STEP 7 ab Version 4.0.

Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

Ihren Ansprechpartner finden Sie unter:

`http://www.siemens.com/automation/partner`

Den Wegweiser zum Angebot an technischen Dokumentationen für die einzelnen SIMATIC Produkte und Systeme finden Sie unter:

`http://www.siemens.de/simatic-tech-doku-portal`

Den Online-Katalog und das Online-Bestellsystem finden Sie unter:

`http://mall.automation.siemens.com`

Trainingscenter

Um Ihnen den Einstieg in das xxx und das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in

D 90327 Nürnberg.
Telefon: +49 (911) 895-3200
Internet: `http://www.sitrain.com`

Technical Support

Sie erreichen den Technical Support für alle A&D-Produkte

- Über das Web-Formular für den Support Request `http://www.siemens.de/automation/support-request`
- Telefon: + 49 180 5050 222.
- Fax:+ 49 180 5050 223

Weitere Informationen zu unserem Technical Support finden Sie im Internet unter `http://www.siemens.com/automation/service`.

Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser komplettes Wissen online an.

`http://www.siemens.com/automation/service&support`

Dort finden Sie:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche in Service & Support.
- ein Forum, in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Automation & Drives vor Ort.
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile. Vieles mehr steht für Sie unter dem Begriff ”Leistungen” bereit.

Inhaltsverzeichnis

Teil 1: Planung des Umstiegs

Einleitung 1

Sie kennen bisher den Namen SIMATIC als Synonym für unsere Steuerung - die S5. Heute jedoch steht der Namen SIMATIC für Vollintegrierte Automation.

Der Begriff **Vollintegrierte Automation** beschreibt eine revolutionäre neue Art und Weise, die Welt der Fertigungs- und Prozeßtechnik zu vereinen. Sämtliche Hard- und Softwarekomponenten sind dabei in einem einzigen System integriert: SIMATIC.

Möglich wird diese vollständige Integration durch die dreifache Durchgängigkeit, die wir mit dem System 7 bieten:

- In der Datenhaltung

  Daten werden nur noch einmal eingegeben und stehen fabrikweit zur Verfügung. Übertragungsfehler und Inkonsistenzen gehören damit der Vergangenheit an.

- In der Projektierung und Programmierung

  Sämtliche zu einer Lösung gehörenden Komponenten und Systeme werden mit einem einzigen vollintegrierten und dabei modular aufgebauten Software-Baukasten projektiert, konfiguriert, programmiert, in Betrieb genommen, getestet und überwacht - unter einer Bedienoberfläche und mit dem genau passenden Werkzeug.

- In der Kommunikation

  Das ”Wer mit wem” wird einfach über eine Verbindungstabelle festgelegt und kann jederzeit an jeder Stelle geändert werden. Die unterschiedlichen Netze lassen sich einfach und einheitlich projektieren.

Um diesem neuen Verständnis der SIMATIC als vollintegriertes System gerecht zu werden, sind in der SIMATIC S7 neueste Konzepte zum tragen gekommen. Dadurch wurden Funktionen teilweise anders realisiert, als Sie es von der S5 her kennen.

Auch bei der Programmiersoftware STEP 7 haben wir auf neueste Technologien und Konzepte gesetzt. So ist z.B. die Bedienoberfläche vollständig unter Windows 95/NT nach modernen ergonomischen Erkenntnissen gestaltet. Bei den Programmiersprachen haben wir viel Wert darauf gelegt, möglichst weitgehend die IEC 1131 Norm zu unterstützen ohne auf die Kompatibilität zu STEP 5 zu verzichten.

Wir sind zuversichtlich, daß wir den daraus resultierenden Anforderungen an STEP 7

- Software-Basis für die vollintegrierte Automatisierung
- IEC 1131 konforme Programmierung
- Kompatibilität zu STEP 5

weitestgehend gerecht geworden sind.

Wir wissen aber auch, daß der Umstieg von einem bestehenden System auf ein neues System Fragen aufwirft und speziell im Bereich der Software die Notwendigkeit besteht, gewisse Anpassungen vorzunehmen.

Die vorliegende Broschüre soll Ihnen eine Antwort auf diesen Fragen geben und gleichzeitig einen einfachen Weg aufzeigen, wie Sie ihre bestehenden STEP 5 Programme in der SIMATIC S7 weiter verwenden können.

2 Hardware

In diesem Kapitel wird die in S7 einsetzbare Hardware beschrieben und bei Bedarf mit der Hardware bei S5 verglichen, um Ihnen den Übergang von S5 nach S7 zu erleichtern.

Hardware-Umsetzung S5 > S7 mit Siemens-Katalog auf CD-ROM

Für die hardwaremäßige Umsetzung von S5 nach S7 steht Ihnen auf der CD-ROM ”Komponenten für die Automation” / Katalog CA01 (ab 04/97) eine Applikation zur Verfügung, die Sie im Produktkatalog unter dem Menüpunkt **Auswahlhilfen > Simatic** finden. Hier können Sie eine beliebige S5-Anlage eingeben. Die Applikation erzeugt aus den vorgegebenen Daten einen Rackaufbau und eine Signalliste. Die erzeugte Konfiguration können Sie dann in einen S7-Aufbau umsetzen lassen.

2.1 Automatisierungssysteme

SIMATIC S7 besteht aus drei im Leistungsspektrum abgestuften Automatisierungssystemen.

SIMATIC S7-200

SIMATIC S7-200 ist eine kompakte Micro-SPS für den untersten Leistungsbereich. Für diese Steuerung gibt es ein S7-200-systemspezifisches Software-Paket, das in die nachfolgende Umstiegshilfe S5 - S7 nicht miteinbezogen wurde, da eine solche aufgrund der Systemeigenschaften von S7-200 nicht softwaregestützt durchgeführt werden kann.

SIMATIC S7-300

SIMATIC S7-300 ist eine modulare Kleinsteuerung für den unteren Leistungsbereich.

SIMATIC S7-400

SIMATIC S7-400 deckt den mittleren, oberen und obersten Leistungsbereich ab.

Zur besseren Orientierung beginnen die Bezeichnungen von S7-300-Baugruppen immer mit 3 und die von S7-400-Baugruppen mit 4.

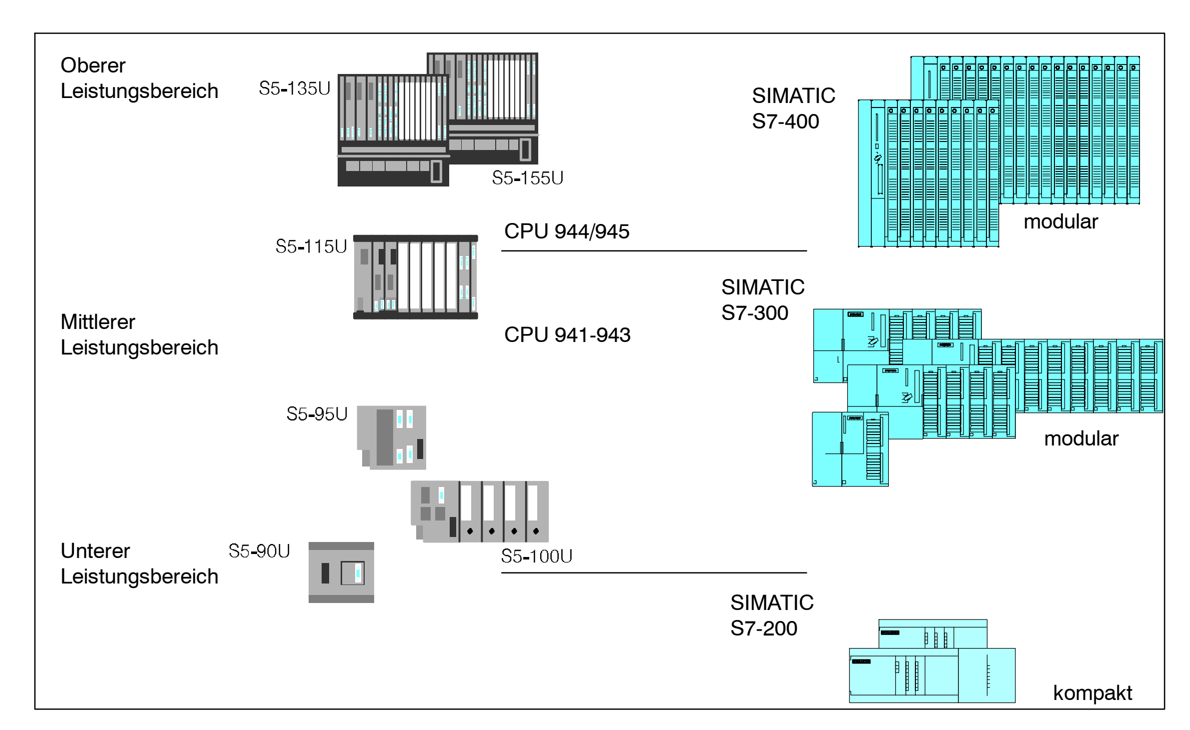


Bild 2-1 Automatisierungssysteme SIMATIC

Anschluß von PG und OP an SIMATIC S7

Programmiergeräteschnittstelle MPI (Multi Point Interface) für PG und OP

Die Programmiergeräteschnittstelle AS511 der SIMATIC S5 wurde durch die Mehrpunktfähige Schnittstelle MPI (für S7-300 und S7-400) ersetzt. Die MPI-Schnittstelle dient dem direkten elektrischen Anschluß der HMI-Geräte (HMI: Human Machine Interface, früher COROS) und der Programmiergeräte an die Programmiergeräte-Schnittstelle der SIMATIC S7. Die Schnittstellen sind fest integriert.

In der folgenden Tabelle werden die Schnittstellen in einer Gegenüberstellung erläutert.

AS511	MPI
25polige TTY-Schnittstelle (20 mA)	9polige Sub-D-Schnittstelle mit RS485-Technik
Baudrate: 9,6 kBaud	Baudrate: 187,5 kBaud
Protokoll: 3964R	Protokoll: S7-Funktionen
	Netzausdehnung: 50 m mit Busverstärkern oder speziellen Kabeln bis über 1000m Alle programmierbaren Baugruppen innerhalb eines Aufbaus über MPI ansprechbar
Ein Gerät anschließbar	Bis zu 31 Geräte anschließbar

Busschnittstelle für OP

Über das Bussystem **PROFIBUS** (frühere Bezeichnung: SINEC L2) können Automatisierungsgeräte/-systeme der Automatisierungsfamilien SIMATIC S5 und SIMATIC S7 angeschlossen werden. Der Anschluß erfolgt busspezifisch wie bisher.

2.2 S7-Baugruppen

Keine grundsätzlichen Änderungen gegenüber S5

Das Baugruppenspektrum der S7 entspricht dem von SIMATIC S5 bekannten und bewährten Baugruppenkonzept und wird dieses weiter ausbauen und ergänzen.

Es gibt bei S7 folgende Baugruppentypen:

- Zentralbaugruppen (CPU),
- Stromversorgungsbaugruppen (PS),
- Anschaltungsbaugruppen (IM),
- Kommunikationsbaugruppen CP; (z. B. für Anschluß an PROFIBUS),
- Funktionsbaugruppen FM; (z. B. zum Zählen, Positionieren, Regeln),
- Digital- und Analogbaugruppen heißen jetzt Signalbaugruppen (SM).

In diesem Kapitel erfahren Sie, wo die Gemeinsamkeiten und Unterschiede im Baugruppenspektrum von SIMATIC S5 und SIMATIC S7 liegen.

Neue Leistungsmerkmale

STEP 7-Baugruppen zeichnen sich durch diese neuen Leistungsmerkmale aus:

- Es sind keine Brücken und Schalter mehr auf den Baugruppen vorhanden.
- Alle Baugruppen sind lüfterlos betreibbar. Sie haben wie bei S5 die Schutzart IP 20.
- Sie haben parametrierbare und diagnosefähige Baugruppen zur Auswahl!
- Die Steckplatzbelegung bei S7 kann flexibler erfolgen als bei S5.
- Erweiterungsgeräte und Dezentrale Peripheriegeräte ET 200 können Alarme auslösen.

Vergleich der Baugruppenparametrierung S5/S7

Die folgende Tabelle zeigt eine Vergleich der Baugruppenparametrierung in SIMATIC S5 und SIMATIC S7:

SIMATIC S5	SIMATIC S7
	Anordnung der Baugruppen (Konfiguration der Hardware) mit dem STEP 7-Applikation zum Konfigurieren der Hardware
Einstellen der Adressen mit DIL-Schaltern	Adresseneinstellung mit der STEP 7-Applikation zum Konfigurieren der Hardware bzw. steckplatzorientiert
Einstellen des Betriebsverhaltens mit DIL-Schaltern	Parametrieren der Baugruppen mit der STEP 7-Applikation zum Konfigurieren der Hardware
Parametrierung des Betriebsverhaltens der Zentralbaugruppen über Systemdatenbereiche bzw. DB 1 / DX 0	Parametrieren der CPU mit der STEP 7-Applikation zum Konfigurieren der Hardware
	Übertragen der übersetzten Konfigurationsdaten zur CPU; automatisches Übertragen der Baugruppenparameter im Anlauf

2.2.1 Zentralbaugruppen (CPU)

CPUs der S7-300

Die Tabelle 2-1 enthält die wichtigsten Leistungsmerkmale der verschiedenen CPUs der S7-300. Möchten Sie eine S5-CPU ersetzen, können Sie zur Wahl der geeigneten CPU die Leistungsmerkmale vergleichen.

Tabelle 2-1 Leistungsmerkmale der S7-300 CPUs

Leistungsmerkmal	312 IFM	313	314	314 IFM	315	315-2 DP
Arbeitsspeicher (integriert)	6 kByte	12 kByte	24 kByte	24 kByte	48 kByte	
Ladespeicher • integriert	20 kByte RAM; 20 kByte EEPROM	20 kByte RAM	40 kByte RAM	40 kByte RAM; 40 kByte EEPROM	80 kByte RAM	
• erweiterbar mit Memory Card	–	bis 512 kByte	bis 512 kByte	–	bis 512 kByte (in CPU programmierbar bis 256 kByte)	
Größe des Prozeßabbildes, jeweils, Ein- und Ausgänge	32 Byte + 4 onboard	128 Byte	128 Byte	124 Byte + 4 onboard	128 Byte	
Peripherieadreßraum • Digitalein-/ausgänge	Eingänge: 128 + 10 onboard Ausgänge: 128 + 6 onboard	128	512	Eingänge: 496 + 20 onboard Ausgänge: 496 + 16 onboard	1024	
• Analogein-/ausgänge	32		64	Eingänge: 64 + 4 onboard Ausgänge: 64 + 1 onboard	128	
Merker	1024	2048				
Zähler	32	64				
Zeiten	64	128				
maximale Summe aller remanenten Daten	72 Bytes		4736 Bytes	144 Bytes	4736 Bytes	
Lokaldaten	512 Bytes insgesamt; 256 Bytes je Prioritätsklasse	1536 Bytes insgesamt; 256 Bytes je Prioritätsklasse				
Bausteine:						
OBs	3	13	13	13	13	14
FBs	32	128	128	128	128	128
FCs	32	128	128	128	128	128
DBs	63	127	127	127	127	127
SFCs	25	44	48	48	48	53
SFBs	2	7	7	14	7	7

CPUs der S7-400

Die CPUs der S7-400 unterscheiden sich in ihrem Leistungsumfang. Tabelle 2-2 zeigt eine Gegenüberstellung der Leistungsmerkmale dieser CPUs.

Tabelle 2-2 Leistungsmerkmale der CPUs der S7-400

Leistungsmerkmal	CPU 412-1	CPU 413-1	CPU 413-2 DP	CPU 414-1	CPU 414-2 DP	CPU 416-1	CPU 416-2 DP
Arbeitsspeicher integriert	48 kByte	72 kByte		128 kByte	128/384 kByte	512 kByte	0,8/1,6 MByte
Ladespeicher • integriert • erweiterbar mit Memory Card	8 kByte bis 15 MByte			8 kByte bis 15 MByte		16 kByte bis 15 MByte	
Größe des Prozeßabbildes, jeweils Eingänge und Ausgänge	128 Byte			256 Byte		512 Byte	
Peripherieadreßraum • Digitalein/ausgänge max. • Analogein/ausgänge max.	2 kByte 16384 1024			8 kByte 65536 4096		16 kByte 131072 8192	
Merker	4096 M 0.0 bis M 511.7			8192 M 0.0 bis M 1023.7		16384 M 0.0 bis M 2047.7	
Zähler	256 Z 0 bis Z 255			256 Z 0 bis Z 255		512 Z 0 bis Z 511	
Zeiten	256 T 0 bis T 255			256 T 0 bis T 255		512 T 0 bis T 511	
Lokaldaten	4 KByte insgesamt			8 KByte insgesamt		16 KByte insgesamt	
Bausteine: OBs FBs FCs DBs SFBs SDBs	 23 256 256 511 24 512			 31 512 1024 1023 24 512		 44 2048 2048 4095 24 512	
SFCs	55	55	58	55	58	55	58

Remanenz bei S7-400

Die Zentralbaugruppen der SIMATIC S7-400 benötigen zur Pufferung von Zeiten, Zählern und Merkern eine Pufferbatterie.

Remanenz ohne Batterie bei S7-300

Zur Pufferung von Zeiten, Zählern und Merkern auf der Zentralbaugruppe benötigen Sie bei S7-300 keine Batterien. Ebenso können Sie bei S7-300 den Inhalt von Datenbausteinen netzausfallsicher halten. Die Zentralbaugruppen der SIMATIC S7-300 haben einen wartungsfreien Backup-Puffer, der bei Netzausfall die als remanent parametrierten Operanden und Daten speichert.

Die Anzahl und die Größe der Remanenzbereiche sind CPU-abhängig.

Parametrieren der Remanenz

Das Festlegen des remanent zu haltenden Bereiches erfolgt mittels Parametrierdialoge im Rahmen der Hardwareprojektierung mit STEP 7.

2.2.2 Stromversorgungsbaugruppen (PS)

Für jedes Automatisierungssystem stehen Ihnen verschiedene Stromversorgungsbaugruppen zur Verfügung.

Stromversorgungsbaugruppe in S7-300

Für die Stromversorgung der CPU in S7-300 kann jedes 24V-Netz (Industrie) verwendet werden.

Im Baugruppenspektrum der S7 sind folgende Stromversorgungsbaugruppen enthalten, die speziell auf S7-300 abgestimmt sind:

Bezeichnung	Ausgangs-strom	Ausgangs-spannung	Eingangs-spannung
PS 307	2A	DC 24V	AC 120V / 230V
PS 307	5A	DC 24V	AC 120V / 230V
PS 307	10A	DC 24V	AC 120V / 230V

Stromversorgungsbaugruppen in S7-400

Bezeichnung	Ausgangs-strom	Ausgangs-spannung	Eingangs-spannung
PS 407 4A	4A 0,5A	DC 5V DC 24V	AC 120V / 230V
PS 407 10A	10A 1A	DC 5V DC 24V	AC 120V / 230V
PS 407 20A	20A 1A	DC 5V DC 24V	AC 120V / 230V
PS 405 4A	4A 0,5A	DC 5V DC 24V	DC 24V
PS 405 10A	10A 1A	DC 5V DC 24V	DC 24V
PS 405 20A	20A 1A	DC 5V DC 24V	DC 24V

Weitere Informationen sind in den Referenzhandbüchern **/71/** und **/101/** enthalten.

2.2.3 Anschaltungsbaugruppen (IM)

Für einige Anschaltungsbaugruppen, die in der S5 zur Verfügung stehen, gibt es Ersatz in S7. Dies bezieht sich hauptsächlich auf Kopplung im Nahbereich. Für die Kopplung im Fernbereich wird bei S7 empfohlen, die Signale über PROFIBUS zu übermitteln.

Vergleich der IM-Baugruppen

Baugruppe S5	Baugruppe S7-300	Baugruppe S7-400	Beschreibung
IM 305 IM 306 IM 300 / IM 312	IM 365 IM 360 / IM 361	IM 460-0 / IM 461-0 IM 460-1 / IM 461-1	zentraler Aufbau
-	-	IM 460-3 / IM 461-3	Fernbereich (bis 100 m)
IM 301 / IM 310	Kopplung über PROFIBUS	Kopplung über PROFIBUS	Kopplung von Peripherie- und signalvorverarbeitenden Baugruppen (bis 200 m)
IM 304 / IM 314	Kopplung über PROFIBUS	Kopplung über PROFIBUS	Verwendung von Dezentraler Peripherie im Fernbereich (bis 600 m)
	–	IM 463-2	dezentrale Kopplung von S5-Erweiterungsgeräten im Fernbereich (bis 600 m)
IM 307 / IM 317	Kopplung über PROFIBUS	Kopplung über PROFIBUS	Kopplung über Lichtwellenleiter (bis 1500 m)
IM 308 / IM 318	Kopplung über PROFIBUS	Kopplung über PROFIBUS	Entfernung bis 3000 m

Für die Anschaltungsbaugruppe IM 308C kann in S7 ersatzweise die IM 467 eingesetzt werden.

S5-Digital- und Analogbaugruppen können Sie mit der Anschaltungsbaugruppe IM 463-2 über einen S5-Erweiterungsbaugruppenträger mit IM 314 an den S7-Baugruppenträger koppeln.

Anschließbare S5-Erweiterungs-baugruppenträger

Folgende S5-Erweiterungsbaugruppenträger sind anschließbar:

- EG 183
- EG 185
- ER 701-2
- ER 701-3

2.2.4 Kommunikationsbaugruppen (CP)

Nachfolgend werden die S5- und S7-Kommunikationsbaugruppen aufgelistet, die für die verschiedenen Subnetze verwendet werden können. Zusätzlich sind die Dienste angegeben, die durch die Baugruppen unterstützt werden.

Subnetze in der SIMATIC

Den differenzierten Anforderungen der Automatisierungsebenen (Leit-, Zellen-, Feld- und Aktor-Sensor-Ebene) entsprechend, bietet SIMATIC die folgenden Subnetze an:

- **AS-Interface**

  Das AS-Interface oder Aktor-/Sensor-Interface ist ein Verbindungssystem für die unterste Prozeßebene in Automatisierungsanlagen. Es dient speziell zur Vernetzung binärer Sensoren und Aktoren. Die Datenmenge beträgt maximal 4 Bit pro Slave.

- **MPI**

  Das MPI-Subnetz ist für die Feldebene und Zellenebene mit kleinen Ausdehnungen. Die MPI ist eine mehrpunktfähige Schnittstelle in der SIMATIC S7/M7 und C7. Sie ist als PG-Schnittstelle konzipiert und für die Vernetzung weniger CPUs, beim Austausch kleiner Datenmengen (bis 70 Byte), gedacht.

- **PROFIBUS**

  PROFIBUS ist im offenen, herstellerunabhängigen Kommunikationssystem der SIMATIC das Netz für den Zellen- und Feldbereich. Der PROFIBUS ist für die schnelle Übertragung bei mittleren Datenmengen (ca. 200 Byte) geeignet.

- **Industrial Ethernet**

  Industrial Ethernet ist im offenen, herstellerunabhängigen Kommunikationssystem der SIMATIC das Netz für die Leitebene und die Zellenebene. Das Industrial Ethernet ist für schnelle Übertragung bei großen Datenmengen geeignet.

- **Punkt-zu-Punkt-Kopplung**

  Eine Punkt-zu-Punkt-Kopplung ist kein Subnetz im herkömmlichen Sinne. In der SIMATIC wird diese Kopplung mittels Punkt-zu-Punkt Kommunikationsprozessoren (CP) realisiert, wobei zwei Kommunikationspartner (SPS, Scanner, PC, usw.) miteinander verbunden sind.

AS-Interface (SINEC S1)

In der nachfolgenden Tabelle erhalten Sie einen Überblick, welche Baugruppen Ihnen für die Kommunikation über das AS-Interface zur Verfügung stehen.

Baugruppe S5	Baugruppe S7-300	Baugruppe S7-400
CP 2433 (AS-i-Funktionen) CP 2430 (AS-i-Funktionen)	CP 342-2 (AS-i-Funktionen)	-

MPI (SINEC L1)

Die Kommunikation über SINEC L1 bei S5 wurde in S7 durch die Globaldatenkommunikation mit MPI umgesetzt.

Alle CPUs in S7-300 und S7-400, die PGs und OPs besitzen eine MPI-Schnittstelle.

PROFIBUS (SINEC L2)

In der nachfolgenden Tabelle erhalten Sie einen Überblick, welche Baugruppen Ihnen für die Kommunikation mit PROFIBUS zur Verfügung stehen und welche Dienste durch diese Baugruppen unterstützt werden.

Baugruppe S5	Baugruppe S7-300	Baugruppe S7-400
CP5431 (FMS, FDL, DP) CPU 95U (FDL, DP *))	CP 342-5 (S7-Funktionen, FDL, DP) CP 343-5 (S7-Funktionen, FDL, FMS)	CP 443-5 Ext. (S7-Funktionen, FDL, DP) CP 443-5 Basic (S7-Funktionen, FDL, FMS)
IM 308-B/C (DP)	CPU 315-2 DP (DP)	CPU 413-2 DP (DP) CPU 414-2 DP (DP) CPU 416-2 DP (DP) IM 467 (DP)

*) abhängig von der bestellten Ausführung

Industrial Ethernet (SINEC H1)

In der nachfolgenden Tabelle erhalten Sie einen Überblick, welche Baugruppen Ihnen für die Kommunikation mit Industrial Ethernet zur Verfügung stehen und welche Dienste durch diese Baugruppen unterstützt werden.

Baugruppe S5	Baugruppe S7-300	Baugruppe S7-400
CP1430 TF (ISO-Transport)	CP 343-1 (S7-Funktionen, ISO-Transport)	CP 443-1 (S7-Funktionen, ISO-Transport)
CP 1430 TCP (ISO on TCP)	CP 343-1 TCP (S7-Funktionen, ISO on TCP)	CP 443-1 TCP (S7-Funktionen, ISO on TCP)

Punkt-zu-Punkt-Kopplung

In der nachfolgenden Tabelle erhalten Sie einen Überblick, welche Baugruppen Ihnen für die Punkt-zu-Punkt-Kopplung zur Verfügung stehen und welche Dienste durch diese Baugruppen unterstützt werden.

Baugruppe S5	Baugruppe S7-300	Baugruppe S7-400
CP 521 (3964 (R), ASCII) CP 523 (3964 (R), ASCII)	CP 340-RS 232C (3964 (R), ASCII) CP 340-20 mA (3964 (R), ASCII) CP 340-RS 422/485 (3964 (R), ASCII)	CP 441-1 (3964 (R), RK512, ASCII)
CP 544 (3964 (R), RK 512, ASCII)	-	
CP 524/525 (3964 (R), RK 512, ASCII, nachladbare Sondertreiber) CP 544 B (3964 (R), RK 512, ASCII, nachladbare Sondertreiber)	-	CP 441-2 (3964 (R), RK512, ASCII, nachladbare Sondertreiber)

2.2.5 Funktionsbaugruppen (FM)

Für einige IP- und WF-Baugruppen der SIMATIC S5 gibt es die Möglichkeit, sie mit Hilfe der Adaptionskapsel in S7-400 einzusetzen. Ansonsten stehen Ihnen neue Funktionsbaugruppen für S7 zur Auswahl, um die gewünschte Funktionalität zu erfüllen.

Die folgende Tabelle gibt Ihnen einen Überblick über signalvorverarbeitende Baugruppen in S5 und S7.

Tabelle 2-3 Vergleich signalvorverarbeitender Baugruppen in S5 und S7

S5-Baugruppe	Adaptions-kapsel	S7-Baugruppe	Beschreibung
IP 240	ja	FM 451 (bedingt)	Zähler-, Wegerfassungs-, Positionierbaugruppe
IP 241	nein	FM 451 / FM 452 (bedingt)	Digitale Wegerfassungsbaugruppe
IP 242A	nein	nein	Zählerbaugruppe
IP 242B	ja	nein	Zählerbaugruppe
IP 244	ja	FM 455	Reglerbaugruppe
IP 246I/A	ja	FM 354 / FM 357 / FM 453	Positionierbaugruppe für drehzahlregelbare Antriebe
IP 247	ja	FM 353 / FM 357 / FM 453	Positionierbaugruppe für Schrittmotoren
IP 252	nein	FM 455 (bedingt)	Regelungsbaugruppe
IP 260	nein	FM 355 (bedingt)	Regelungsbaugruppe
IP 261	nein	nein	Dosierbaugruppe
IP 281	nein	FM 350-1 / FM 450-1	Zählerbaugruppe
IP 288	nein	FM 451 / FM 452	Positionierbaugruppe für Eil-/Schleichganggetriebe und Nockensteuerung
WF 705	ja	FM 451 (bedingt)	Wegerfassungsbaugruppe
WF 706	nein	FM 451 (bedingt)	Positionier- und Zählbaugruppe
WF 707	nein	FM 452 (bedingt)	Nockensteuerwerk
WF 721	ja	FM 354 (bedingt wegen Aufbautechnik)	Positionierbaugruppe
WF 723A	ja	FM 453	Positionierbaugruppe

Tabelle 2-3 Vergleich signalvorverarbeitender Baugruppen in S5 und S7, Fortsetzung

S5-Baugruppe	Adaptions-kapsel	S7-Baugruppe	Beschreibung
WF 723 B	ja	FM 357 (bedingt wegen Aufbautechnik)	Positionierbaugruppe
WF 723 C	ja	nein	Positionierbaugruppe
-	-	FM 456-4	Applikationsbaugruppe (M7-FM)
-	-	SINUMERIK FM-NC	NC-Steuerung
-	-	FM STEPDRIVE	Ansteuerung von Schrittmotoren
-	-	SIMOSTEP	Schrittmotor

2.2.6 Signalbaugruppen (SM)

Die Signalbaugruppen in SIMATIC S7 sind vergleichbar mit den Ein- / Ausgabebaugruppen, die es in S5 gibt. Es stehen Ihnen jedoch neben den einfachen Signalbaugruppen auch parametrierbare und diagnosefähige Baugruppen zur Auswahl.

Parametrierbare SMs

Bei parametrierbaren Digitaleingabebaugruppen haben Sie z. B. die Möglichkeit, mit der STEP 7-Applikation zum Konfigurieren der Hardware einzustellen, welche Kanäle bei Flankenwechsel einen Prozeßalarm auslösen sollen.

Die Eingangsbereiche von Analogeingabebaugruppen lassen sich sehr einfach mit STEP 7 parametrieren.

Diagnosefähige SMs

Diagnosefähige Baugruppen erkennen sowohl externe Fehler wie z. B. Drahtbruch oder externen Kurzschluß, als auch interne Fehler wie z. B. RAM-Fehler oder internen Baugruppenkurzschluß.

Ein Diagnoseereignis wird auf zwei Arten von der Steuerung ”verarbeitet”:

- Auslösen eines Diagnosealarms (dann wird ein entsprechender OB im Anwenderprogramm aufgerufen, der das zyklische Programm unterbricht)
- Eintrag in den Diagnosepuffer der CPU (mit PG/B+B-Gerät auslesbar)

Die folgenden Tabellen listen die in S7 vorhandenen Signalbaugruppen auf:

Tabelle 2-4 Signalbaugruppen in SIMATIC S7-300

DI (SM 321)	DO (SM 322)	AI (SM 331)	AO (SM 332)
32 x DC 24V	32 x DC 24V/0,5A	8 x 12 Bit	2 x 12 Bit
16 x DC 24V	16 x DC 24V/0,5A	2 x 12 Bit	
16 x DC 24V mit Prozeß- und Diagnosealarm	8 x DC 24V/0,5A mit Diagnosealarm	Ex: 4 x 15 Bit	Ex: 4 x 15 Bit
16 x DC 24V M-lesend	8 x DC 24V/2A	Ex: 12 x 15 Bit	
8 x AC 120V/230V	8 x AC 120V/230V/ 2A	AI 4/AO 2 X 8/8 Bit (SM 334)	
Ex: 4 x DC 24V	Ex: 4 x DC 15V/ 20mA		
	Ex: 4 x DC 24V/ 20mA		

Tabelle 2-5 Signalbaugruppen in SIMATIC S7-400

DI (SM 421)	**DO (SM 422)**	**AI (SM 431)**	**AO (SM 432)**
32 x DC 24V	32 x DC 24V/0,5A	8 x 13 Bit	8 x 13 Bit
16 x UC 24V/60V mit Prozeß- und Diagnosealarm	16 x DC 24V/2A	8 x 14 Bit (für Temperatur-messung)	
16 x UC 120V/230V	16 x AC 120V/230V /5A	8 x 14 Bit	
32 x UC 120V	16 x AC 120V/230V /2A	16 x 16 Bit	
	16 x UC 30V/230V/ Rel 5A		

2.2.7 Simulatorbaugruppen (S7-300)

Zum Test Ihres Programmes steht Ihnen in S7-300 die Simulatorbaugruppe SM 374 zur Verfügung.

Die Simulatorbaugruppe zeichnet sich durch folgende Eigenschaften aus:

- Simulation von
  - 16 Eingängen oder
  - 16 Ausgängen oder
  - 8 Eingängen und 8 Ausgängen (mit jeweils den gleichen Anfangsadressen!)
- Funktion mit Schraubendreher einstellbar
- Statusanzeigen für Simulation von Ein- oder Ausgängen

2.3 Dezentrale Peripherie

Die Baugruppen für Dezentrale Peripherie des Systems ET 200, die Sie in SIMATIC S5 eingesetzt haben, können Sie auch wieder für SIMATIC S7 verwenden.

Zusätzliche neue ET 200-Baugruppen ergänzen das Baugruppenspektrum.

DP-Master

DP-Master im Dezentralen Peripheriesystem können folgende Baugruppen sein:

- S7-300 mit CPU 315-2 DP oder CP 342-5 als DP-Master
- S7-400 mit CPU 413-2 DP / 414-2 DP / 416-2DP oder CP 443-5 Extended als DP-Master

DP-Slaves

DP-Slaves im Dezentralen Peripheriesystem können z. B. sein:

- Dezentrale Peripheriegeräte ET 200B, ET 200C, ET 200M, ET 200X (bis 12 MBaud) und ET 200U, ET 200L (bis 1,5 MBaud)
- Automatisierungsgeräte/-systeme wie
  - S5-115U, S5-135U oder S5-155U mit IM 308-C als DP-Slave
  - S5-95U mit DP-Slave-Schnittstelle (bis 1,5 MBaud)
  - S7-300 mit CPU 315-2 DP oder CP 342-5 als DP-Slave
  - S7-400 mit CP 443-5 als DP-Slave
- Schnittstelle zum Aktuator-Sensor-Interface mit dem DP/AS-i Link
- Textdisplays und Operator Panels für maschinennahes Bedienen und Beobachten
- Identifikationssysteme MOBY
- Niederspannungs-Schaltgeräte
- Feldgeräte von Siemens oder von anderen Herstellen wie z. B. Antriebe, Ventilinseln etc.

FMS-Master

FMS-Master können sein:

- S7-300 mit CP 343-5 als FMS-Master
- S7-400 mit CP 443-5 Basic als FMS-Master

FMS-Slaves

FMS-Slaves können z. B. die ET 200U oder das Motorschutz- und Steuergerät SIMOCODE sein.

Weitere Informationen entnehmen Sie den Handbüchern oder dem Siemens-Katalog CA01.

2.4 Kommunikation

Dienste und Subnetze

Die Kommunikation in der SIMATIC S7 basiert auf unterschiedlichen Subnetzen, auf denen verschiedene Dienste zur Verfügung gestellt werden.

Dienste	S7-Kommunikationsfunktionen (S7-Funktionen)		
	ISO-Transport ISO-on-TCP	FDL (SDA) FMS DP	GD
Subnetze	Industrial Ethernet	PROFIBUS	MPI

Nachfolgend finden Sie eine Zusammenfassung der in der SIMATIC verwendeten Kommunikationsdienste

S7-Funktionen

Die S7-Funktionen bieten Dienste zur Kommunikation zwischen S7/M7-CPUs, S7-OP/Oss und PCs. Die S7-Funktionen sind in jedem SIMATIC S7/M7-Gerät bereits integriert. Da die S7-Funktionen einem Dienst des ISO-Aplikation Layer entsprechen, sind sie unabhängig vom Subnetz und können auf allen Subnetzen (MPI, PROFIBUS, Industrial Ethernet) verwendet werden.

ISO-Transport

Diese Funktionen dienen der gesicherten Übertragung von Daten von der SIMATIC S7 an die SIMATIC S5.

Sie dienen der Übertragung mittlerer Datenmengen (bis 240 Byte) über offene Kommunikation auf Schicht 4 ISO-Transport nach ISO-Referenzmodell bei Industrial Ethernet.

ISO-on-TCP

Diese Funktionen dienen der gesicherten Übertragung von Daten von der SIMATIC S7 an die SIMATIC S5.

Sie dienen der Übertragung mittlerer Datenmengen (bis 240 Byte) über offene Kommunikation gemäß TCP/IP-Protokoll auf Schicht 4 nach ISO-Referenzmodell bei Industrial Ethernet.

Der ISO-on-TCP-Dienst benötigt den erweiterten RFC1006-Standard.

FDL (SDA)

Diese Funktionen dienen der gesicherten Übertragung von Daten von der SIMATIC S7 an die SIMATIC S5.

Sie dienen der Übertragung mittlerer Datenmengen (bis 240 Byte) über offene Kommunikation auf Schicht 2 Fieldbus Data Link (FDL) nach ISO-Referenzmodell bei Industrial Ethernet.

FMS

PROFIBUS FMS (Fieldbus Message Specification) bietet Dienste für die Übertragung von strukturierten Daten (FMS-Variablen) über statische FMS-Verbindungen an.

Der FMS-Dienst läßt sich in die Schicht 7 des ISO-Referenzmodells einordnen. Er entspricht der europäischen Norm EN 50170 Vol. 2 PROFIBUS und bietet Dienste für die Übertragung strukturierter Daten (Variablen).

DP

PROFIBUS-DP-Dienste bieten die Möglichkeit, mit dezentraler Peripherie transparent zu kommunizieren. Vom Steuerungsprogramm her wird dezentrale Peripherie genauso angesprochen wie zentrale Peripherie.

GD

Die Globale Datenkommunikation ist eine einfache, im Betriebssystem der S7-300/400-CPUs integrierte Kommunikationsmöglichkeit.

Die GD-Kommunikation ermöglicht den zyklischen und bei S7-400 auch ereignisgesteuerten Datenaustausch zwischen CPUs über die MPI-Schnittstelle.

2.4.1 Schnittstelle zum Anwenderprogramm

Die Kommunikationsschnittstelle zum Anwenderprogramm bilden folgende Bausteine:

- SFCs (ohne Verbindungsprojektierung)
- SFBs (mit Verbindungsprojektierung) (nur S7-400)
- ladbare FCs / FBs

Diese Bausteine ersetzen die S5-Hantierungsbausteine. Die Funktionalität ist ähnlich, jedoch mit den STEP 7-Sprachmitteln realisiert. Ein entsprechendes S5-Programm mit Hantierungsfunktionen müssen Sie an die neuen Bausteine für die Kommunikation anpassen.

Netz	**Dienst**	**Schnittstelle im S5 Anwenderprogramm**	**Schnittstelle im S7 Anwenderprogramm**
Punkt-zu-Punkt-Kopplung	-	Hantierungsbausteine *	S7-300: ladbare FBs S7-400: ladbare SFBs
PROFIBUS	FDL (AG - AG) Freier Layer 2 FMS	Hantierungsbausteine * Hantierungsbausteine * Hantierungsbausteine *	ladbare FCs - ladbare FBs
Industrial Ethernet	ISO 4 ISO 4 + AP STF MAP	Hantierungsbausteine * Hantierungsbausteine * Hantierungsbausteine * + ladbare FBs Hantierungsbausteine * + ladbare FBs	ladbare FCs - - ladbare FBs

* Je nach CPU integriert oder ladbar.

2.5 Bedienen & Beobachten

Einführung

Die folgenden Abschnitte geben einen Überblick, in welchem Umfang Operator Panel SIMATIC HMI (HMI: Human Machine Interface, früher: COROS) auch bei SIMATIC S7 eingesetzt werden können.

Operator Panels (OP)

Die Operator Panels SIMATIC HMI bieten B&B-Funktionalität in Verbindung mit SIMATIC S5, SIMATIC S7 und SIMATIC TI (sowie anderen Steuerungen).

STEP 5

Für die Anbindung von **SIMATIC OP an SIMATIC S5** gilt allgemein, daß im AG ein Standard-Funktionsbaustein erforderlich ist, der je angeschlossenem OP aufgerufen werden muß.

Folgende OPs sind in Verbindung mit S5 einsetzbar:

- TD17, OP5/A1, OP7/PP, OP7/DP-12, OP15/x1, OP17/PP, OP17/DP-12
- OP25, OP35, OP37, TP37

STEP 7

Hinsichtlich der Kopplung **von SIMATIC OP an SIMATIC S7/M7** ist zwischen PPI, MPI und PROFIBUS (als MPI-Teilnehmer) zu unterscheiden.

PPI bzw. MPI-Kopplung laufen über die PG-Schnittstelle der CPU. Dabei nutzen SIMATIC OPs die Kommunikationsdienste der SIMATIC S7/M7 (S7-Funktionen); ein Standardfunktionsbaustein ist daher nicht erforderlich!

Auch bei der PROFIBUS-Kopplung von SIMATIC OP an SIMATIC S7/M7 handelt es sich um eine Kommunikation auf Basis der S7-Funktionen; ein Standard-FB ist nicht erforderlich! (SIMATIC OPs sind “aktive Teilnehmer” und nicht etwa PROFIBUS-DP /Slaves wie bei der PROFIBUS-Kopplung an SIMATIC S5). Bezüglich Teilnehmeranzahl gilt das gleiche Mengengerüst wie bei einer MPI-Kopplung.

Folgende OPs sind in Verbindung mit S7 einsetzbar:

- TD17, OP3, OP5/A2, OP7/DP, OP7/DP-12, OP15/x2, OP17/DP, OP17/DP-12
- OP25, OP35, OP37, TP37

Aus Sicht der SIMATIC OP gelten folgende Einschränkungen:

- OP3: max 2 Verbindungen
- OP5/15/25: max. 4 Verbindungen
- TD17, OP7/17: max 4 Verbindungen
- OP35: max 6 Verbindungen
- OP37, TP37: max 8 Verbindungen

Projektierung

SIMATIC ProTool und SIMATIC ProTool/Lite sind moderne Projektierungswerkzeuge zur Projektierung der Operator Panels. Während mit SIMATIC ProTool alle Geräte projektiert werden können, ist mit SIMATIC ProTool/Lite die Projektierung auf die zeilenorientierten Operator Panels beschränkt. Funktional ist ProTool/Lite ein Subset von ProTool.

Integration in SIMATIC STEP 7

ProTool ist in die Projektierungssoftware STEP7 der SIMATIC integrierbar und ermöglicht dadurch den direkten Zugriff auf Projektierungsdaten, wie z.B. Symbolliste, Kommunikationsparameter der Steuerungsprojektierung. Dies spart Zeit und Geld; mögliche Fehler durch mehrmaliges Eingeben werden vermieden.

Tabelle 2-6 Projektierungstools für Bedien- und Beobachtungsgeräte

Gerät	Projektierungstool
zeilenorientierte OP (TD17, OP 3, OP 5, OP7,OP 15, OP17)	ProTool/Lite bzw. ProTool
grafikorientierte OP (OP 25, OP 35, OP37, TP37)	ProTool

WinCC

WinCC kann als Einzelplatz- oder Mehrplatzsystem (Client-Server-Struktur) eingesetzt werden.

WinCC ist ein branchen- und technologieneutrales System zur Lösung von Visualisierungs- und leittechnischen Aufgaben in der Produktions- und Prozeßautomatisierung. Es bietet industriegerechte Funktionsmodule zur Grafikdarstellung, zum Melden, Archivieren und Protokollieren. Mit seiner leistungsfähigen Prozeßkopplung, der schnellen Bildaktualisierung und der sicheren Datenarchivierung gewährleistet es eine hohe Verfügbarkeit.

Neben diesen Systemfunktionen bietet WinCC offene Schnittstellen für Anwenderlösungen. Sie ermöglichen die Integration von WinCC in komplexe, unternehmensweite Automatisierungslösungen. Integriert ist der Archivdatenzugriff über ODBC und SQL, d. h. Standardschnittstellen sowie die Einbindung von Objekten und Dokumenten über OLE2.0 und OLE-Custom-Controls (OCX). Diese Mechanismen machen WinCC zu einem kompetenten, kommunikativen Partner in der Windows-Welt.

Basis für WinCC sind die 32-bit-Betriebssysteme MS-Windows 95 oder MS-Windows NT. Beide verfügen über ein preemtives Multitasting, das schnelle Reaktionen auf Prozeßereignisse und Sicherheit vor Datenverlust in hohem Maße gewährleistet. Windows NT bietet zusätzlich sicherheitsgerichtete Funktionen und dient als Basis für den Serverbetrieb in einem WinCC-Mehrplatzsystem. Die WinCC-Software selbst ist eine 32-bit-Applikation, die mit modernster, objektorientierter Softwaretechnik entwickelt wurde.

3 Software

3.1 Allgemeine Bedienphilosophie

Übersicht

Die Software zur Konfigurierung und Programmierung von SIMATIC S7/M7/C7 ist nach modernen ergonomischen Erkenntnissen gestaltet und deshalb weitgehend selbsterklärend.

3.1.1 Installationsvoraussetzungen

Betriebssystem

Microsoft Windows 95.

Basishardware

Programmiergerät oder PC mit

- Prozessor 80486 (oder höher) und
- RAM-Speicherausbau: mindestens 16 MB, empfohlen sind 32 MB,
- einem VGA-Monitor oder einem anderen Monitor, der von Microsoft Windows 95 unterstützt wird,
- Tastatur und optional, aber empfehlenswert eine Maus, die von Microsoft Windows 95 unterstützt wird.

Speicherkapazität

Erforderlicher Speicherplatz auf der Festplatte

- 105 MB belegt das Basispaket bei Installation von einer Sprache. Der Speicherplatzbedarf ist abhängig vom gewählten Installationsumfang der Basissoftware.
- Etwa 64 MB abzüglich Hauptspeicherausbau sollte STEP 7 zum Anlegen von Swap-Dateien zur Verfügung haben (d. h. ca. 32 MB bei einem Hauptspeicherausbau von 32 MB).
- Ca. 50 MB sollten Sie für Ihre Anwenderdaten vorsehen.
- Mindestens 1 MB freien Speicherplatz auf dem Laufwerk C: für das Setup (Setup-Dateien werden nach Abschluß der Installation gelöscht).

3.1.2 Installieren der STEP 7-Software

Übersicht

STEP 7 enthält ein Setup-Programm, das die Installation automatisch durchführt. Eingabeaufforderungen auf dem Bildschirm führen Sie Schritt für Schritt durch den gesamten Installationsvorgang.

Autorisierung

Für die Nutzung der STEP 7-Programmiersoftware wird eine produktspezifische Autorisierung (Nutzungsberechtigung) benötigt. Die so geschützte Software ist nur benutzbar, wenn auf der Festplatte des betreffenden PG / PC die für das Programm oder Softwarepaket erforderliche Autorisierung erkannt wird.

Für die Autorisierung benötigen Sie die zum Lieferumfang gehörende kopiergeschützte Autorisierungsdiskette. Sie enthält die Autorisierung und das zum Anzeigen, Installieren und Deinstallieren der Autorisisierung erforderliche Programm AUTHORS.

Die Vorgehensweise beim Übertragen und Entfernen der Autorisierung ist im Benutzerhandbuch **/231/** beschrieben.

Hinweis

Siemens Programmiergeräte (wie zum Beispiel das PG 740) werden mit installierbarer STEP 7-Software auf der Festplatte ausgeliefert.

Weitere Informationen zur Installation entnehmen Sie bitte dem Benutzerhandbuch **/231/**.

3.1.3 Starten der STEP 7-Software

Starten

Nach dem Start von Windows 95/NT finden Sie auf der Windows-Oberfläche ein Symbol für den SIMATIC Manager, dem Einstieg in die STEP 7-Software.

STEP 7 wird am schnellsten mit einem Doppelklick auf das Symbol "SIMATIC Manager" gestartet. Daraufhin wird das Fenster des SIMATIC Managers geöffnet. Von hier aus lassen sich alle von Ihnen installierten Funktionen sowohl des Basissystems als auch von Optionssoftware ansprechen.

Alternativ können Sie den SIMATIC Manager auch über die Schaltfläche "Start" auf der Task-Leiste in Windows 95/NT starten: Den Eintrag finden Sie unter "Simatic/STEP 7".

SIMATIC Manager

Der SIMATIC Manager ist die Einstiegsoberfläche für die Konfigurierung und Programmierung. Sie können:

- Projekte einrichten,
- Hardware konfigurieren und parametrieren,
- Kommunikations-Verbindungen projektieren,
- Programme erstellen,
- Ihre Programme testen und in Betrieb nehmen.

Der Zugang zur Funktionalität ist objektorientiert gestaltet und intuitiv und leicht erlernbar.

Sie können mit dem SIMATIC Manager

- offline d.h. ohne angeschlossene Steuerung oder
- online d.h. mit angeschlossener Steuerung

arbeiten. (Beachten Sie dabei die entsprechenden Sicherheitshinweise.)

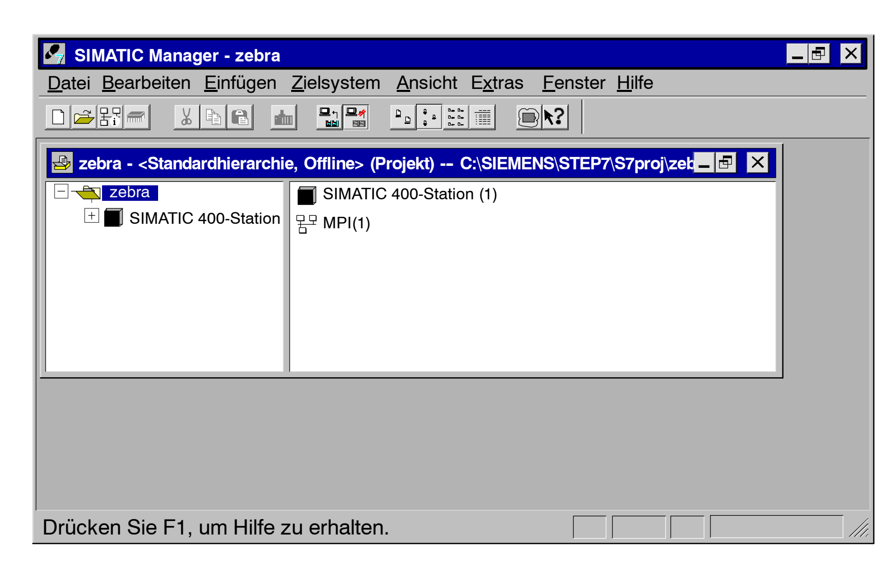


Bild 3-1 Geöffnetes Projekt im SIMATIC Manager

3.2 Aufbau eines S7-Projektes

Definition

Projekte repräsentieren die Gesamtheit aller Daten und Programme einer Automatisierungslösung. Sie dienen dazu, die bei der Erstellung der Automatisierungslösung anfallenden Daten und Programme geordnet abzulegen.

Projekt bei STEP 5

Den Begriff ”Projekt” kennen Sie bereits von STEP 5. Bei STEP 5 faßt ein Projekt alle zu **einem** Anwenderprogramm erstellten **STEP 5**-Dateien in einer Projektdatei zusammen.

In der Projektdatei sind Informationen gespeichert, die eine bequeme Bearbeitung und Pflege eines Anwenderprogramms ermöglichen, z. B. Parametereinstellungen und Katalog-/Dateibezeichnungen.

Projekt bei STEP 7

Ein Projekt umfaßt bei STEP 7 die gesamte Programm- und Datenhaltung für eine Automatisierungslösung, unabhängig von der Anzahl der Zentralbaugruppen und deren Vernetzung. Ein Projekt beschränkt sich also nicht auf ein Anwenderprogramm für eine programmierbare Baugruppe, sondern mehrere Anwenderprogramme für mehrere programmierbare Baugruppen, die unter einem gemeinsamen Projektnamen zusammengefaßt sind.

Anmerkung

Selbstverständlich haben Sie die Möglichkeit, wie von STEP 5 gewohnt, ein einfaches Anwenderprogramm für nur eine Zentralbaugruppe zu erstellen. Ein Projekt ist dann auf eine Zentralbaugruppe beschränkt.

Im folgenden geht es um die Verzeichnisstruktur, die STEP 7 den von Ihnen erstellten Anwenderprogrammen und Daten zur Verfügung stellt.

Bestandteile eines Projekts

Ein Projekt bei STEP 7 umfaßt im wesentlichen die in Bild 3-2 aufgeführten Objekte. Die Objekte sind im Anschluß an das Bild erläutert.

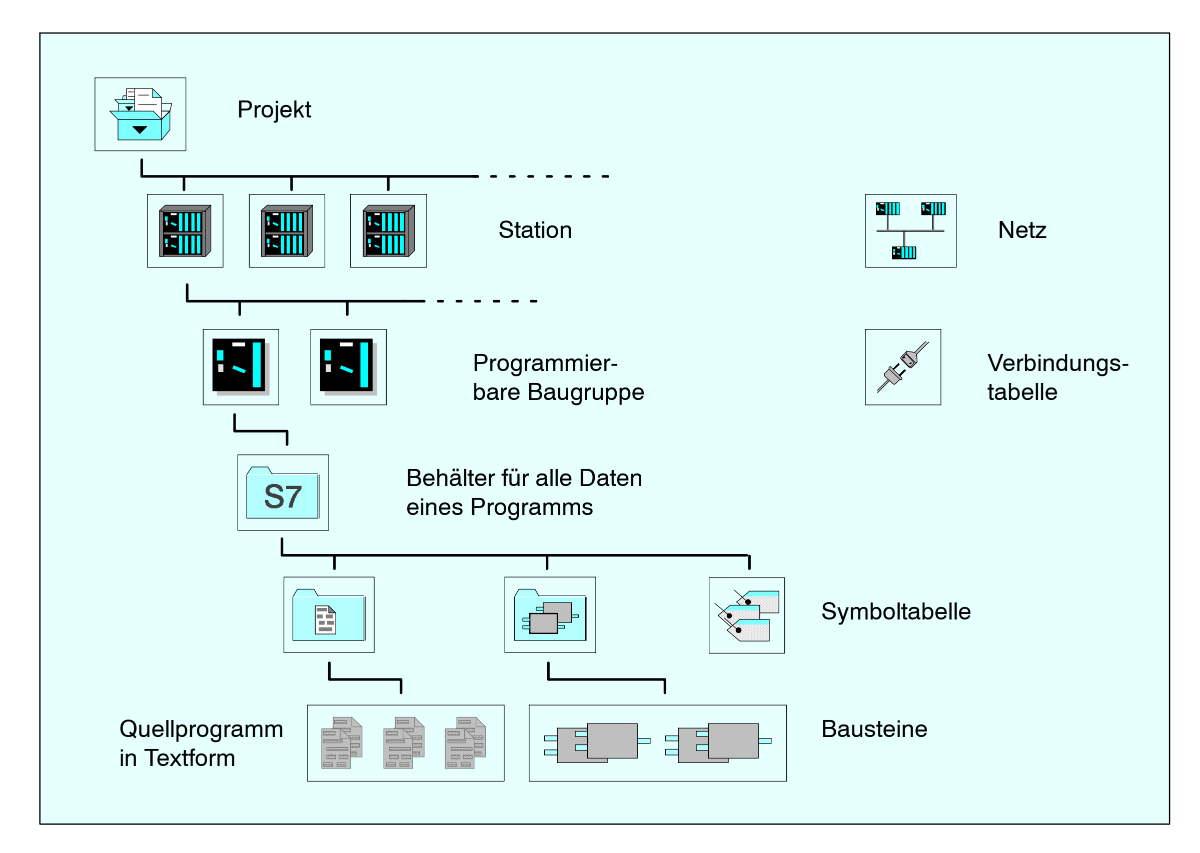


Bild 3-2 Wesentliche Objekte eines Projekts bei STEP 7 und ihre hierarchische Struktur

Netz

Das Objekt ”Netz” repräsentiert die für ein Subnetz (z. B. MPI oder PROFIBUS) maßgeblichen Eigenschaften. Durch die Zuweisung einer Station bzw. einer darin enthaltenen kommunikationsfähigen Baugruppe zu einem Netz kann STEP 7 kommunikationsrelevante Parameter auf Konsistenz prüfen.

Station

Eine Station repräsentiert den Aufbau eines Automatisierungssystems mit allen dazugehörigen Baugruppenträgern. Wenn eine Baugruppe mit DP-Schnittstelle in einer Station steckt, dann ist auch das gesamte Mastersystem (d. h. die dazugehörigen DP-Slaves) ein Teil dieser Station!

Eine Station umfaßt eine oder mehrere programmierbare Baugruppen (z. B. CPU).

Hardware

Hardware ist ein Objekt mit den Konfigurationsdaten und Parametern einer Station. Die Konfigurationsdaten und Parameter einer Station werden in Systemdatenbausteinen (SDBs) abgelegt.

Programmierbare Baugruppe

Programmierbare Baugruppen sind im Gegensatz zu anderen Baugruppen Träger von Anwenderprogrammen. Sie finden in den Ordnern - sie heißen bei STEP 7 ”Behälter” - unter den programmierbaren Baugruppen alle Daten, die zum Programm der Baugruppe gehören:

- Quellprogramme in Textform (die mit einem Texteditor erstellt werden)

  Beim Übersetzen der Quellprogramme werden im Behälter “Bausteine” ablauffähige Bausteine erzeugt.
- Bausteine (die in die programmierbare Baugruppe geladen werden)
- Symboltabelle

Verbindungs-tabelle

Die Verbindungstabelle repräsentiert sämtliche Verbindungen einer programmierbaren Baugruppe (z. B. CPU) in einer Station. Eine Verbindung definiert die Kommunikationseigenschaften zwischen zwei Teilnehmern und wird durch eine Verbindungs-ID gekennzeichnet. Nur diese Verbindungs-ID benötigen Sie, um mit standardisierten Kommunikationsbausteinen - vergleichbar mit den Hantierungsbausteine von STEP 5 - ereignisgesteuerte Kommunikation zu programmieren.

Quellen

Quellen dienen bei der S7-Programmierung als Basis zur Erzeugung von Bausteinen. Quellen können nicht in eine S7-CPU geladen werden.

Bausteine

Bausteine sind durch ihre Funktion, ihre Struktur oder ihren Verwendungszweck abgegrenzte Teile des Anwenderprogramms. Bausteine können in S7-CPUs geladen werden.

Im Behälter “Bausteine” sind neben den ablauffähigen Bausteinen die Variablentabellen enthalten.

Symboltabelle

Die Symboltabelle enhält die Zuweisung von Namen (=Symbolen) z. B. für Eingänge, Ausgänge, Merker und Bausteine.

3.3 Projekt bearbeiten mit SIMATIC-Manager

3.3.1 Projekt anlegen

Neues Projekt

Um ein Projekt anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei ▸ Neu** im SIMATIC Manager.
2. Wählen Sie im Dialogfeld ”Neu” die Option ”Neues Projekt”.
3. Tragen Sie einen Namen für das Projekt ein und bestätigen Sie mit ”OK”.

Alternative für die Weiterarbeit

Bei der Wahl der weiteren Bearbeitungsreihenfolge eines Projekts haben Sie größtmögliche Flexibilität. Nach dem Anlegen eines Projekts können Sie

- Hardware konfigurieren und anschließend die Software dafür erstellen oder
- unabhängig von einer konfigurierten Hardware mit der Software-Erstellung beginnen. Der Hardware-Aufbau einer Station braucht zur Eingabe von Programmen noch nicht festgelegt zu sein.

Tabelle 3-1 Alternativen für die Weiterarbeit

Alternative 1	Alternative 2
Zunächst Hardware konfigurieren (siehe auch *Kap. 3.4*)	**Zunächst Software erstellen**
Konfigurieren Sie Ihre Hardware (siehe Abschnitt 3.4).	
Nach dem Konfigurieren sind die für die Software-Erstellung erforderlichen Behälter ”S7-Programm” bereits eingefügt.	Fügen Sie die erforderlichen Software-Behälter (S7-Programme) in Ihr Projekt ein (siehe Abschnitt 3.6).
Erstellen Sie anschließend die Software für die programmierbaren Baugruppen (siehe Abschnitt 3.6).	Erstellen Sie anschließend die Software für die programmierbaren Baugruppen (siehe Abschnitt 3.6).
	Konfigurieren Sie Ihre Hardware (siehe Abschnitt 3.4).
	Ordnen Sie das S7-Programm einer CPU zu, nachdem Sie die Hardware konfiguriert haben.

Wie Sie Programme auch ohne Hardware-Konfigurierung laden und testen können, ist im Benutzerhandbuch **/231/** beschrieben.

3.3.2 Sichern von Projekten

Übersicht

Um ein Projekt zu sichern, können Sie eine Kopie des Projektes unter einem anderen Namen abspeichern oder das Projekt archivieren.

Speichern unter ...

Gehen Sie folgendermaßen vor:

1. Öffnen Sie das Projekt.
2. Wählen Sie den Menübefehl **Datei ▸ Speichern unter**. Das Dialogfeld ”Speichern unter” wird angezeigt.
3. Wählen Sie Speichern mit oder Speichern ohne Konsistenzüberprüfung und schließen Sie das Dialogfeld mit ”OK”. Das Dialogfeld ”Projekt speichern unter” wird angezeigt.
4. Wählen Sie bei ”Speichern in” das Verzeichnis aus, in dem das Projekt gespeichert werden soll.
5. Geben Sie im Feld ”Dateiname” anstelle des Sterns (*) einen Dateinamen ein. Ändern Sie die Dateierweiterung nicht.
6. Schließen Sie das Dialogfeld mit ”OK”.

Es muß sichergestellt sein, daß auf dem ausgewählten Laufwerk genügend Speicherplatz vorhanden ist. Da in der Regel der Speicherplatz auf einer Diskette zum Speichern eines Projektes unzureichend ist, macht es keinen Sinn, hier ein Diskettenlaufwerk auszuwählen, Um Projekte auf Disketten zu sichern, müssen Sie diese archivieren. Archive können auf mehrere Disketten aufgeteilt werden.

Archivieren

Einzelne Projekte oder Bibliotheken können Sie in komprimierter Form in einer Archivdatei ablegen. Dieses komprimierte Ablegen ist auf Festplatte oder auf transportablen Datenträgern (Diskette) möglich.

Um auf Komponenten eines archivierten Projekts oder einer archivierten Bibliothek zuzugreifen, muß das Projekt zuvor wieder dearchiviert werden. Das Thema Archivieren ist ausführlich im Benutzerhandbuch **/231/** beschrieben.

3.4 Hardware konfigurieren mit STEP 7

Bei SIMATIC S5 gab es keine Möglichkeit, die Hardware per Software zu konfigurieren. In S7 wird das Adressieren und Parametrieren der Baugruppen und die Projektierung der Kommunikation durch eine STEP 7-Applikation übernommen. Dies hat den Vorteil, daß der Anwender an den Baugruppen nichts mehr einstellen muß, da er die Konfigurierung und Parametrierung zentral vom PG aus erledigen kann.

Voraussetzung

Es wurde bereits ein Projekt angelegt.

Einfügen einer Station

Um eine neue Station in einem Projekt anzulegen, öffnen Sie das Projekt, so daß das Projektfenster angezeigt wird (falls noch nicht geschehen).

- Markieren Sie das Projekt.
- Erzeugen Sie das Objekt für die gewünschte Hardware über Menübefehl **Einfügen > Station**.

Sie können im Folgemenü auswählen:

- SIMATIC 300-Station
- SIMATIC 400-Station
- PC/PG
- SIMATIC S5
- Andere Stationen, d. h. nicht SIMATIC S7/M7, SIMATIC S5

Die Stationen PC/PG, SIMATIC S5 und andere Stationen werden nur für die Projektierung von Kommunikationsverbindungen angegeben. Es ist keine Konfiguration und Programmierung von S5 möglich.

Klicken Sie im Projektfenster auf das ”+” vor dem Projektsymbol, falls darunter die Station noch nicht angezeigt wird.

Konfigurierung durchführen

Gehen Sie folgendermaßen vor:

- Klicken Sie auf die neu eingefügte Station. Sie enthält das Objekt ”Hardware”.
- Öffnen Sie das Objekt ”Hardware”. Das Fenster ”HWKonfig” wird aufgeblendet.
- Legen Sie im Fenster ”Hardware konfigurieren” den Aufbau der Station fest. Dazu steht Ihnen ein Baugruppenkatalog zur Verfügung, den Sie mit dem Menübefehl **Ansicht > Katalog** aufblenden können, falls er nicht angezeigt wird.
- Fügen Sie zunächst einen Baugruppenträger (Rack) aus dem Baugruppenkatalog in das leere Fenster ein. Anschließend wählen Sie Baugruppen aus, die Sie auf die Rack-Steckplätze verteilen. Je Station muß mindestens eine CPU konfiguriert sein. Alle Eingaben werden automatisch überprüft und abgewiesen, wenn sie unzulässig sind.

Weitere Informationen zum Konfigurieren finden Sie im Benutzerhandbuch **/231/**.

Ergebnis der Konfigurierung

Nach dem Speichern und Beenden der Hardware-Konfigurierung wird zu jeder CPU, die Sie bei der Konfigurierung angelegt haben, automatisch eine Verbindungstabelle (Objekt ”Verbindungen”) sowie ein S7-Programm angelegt. Das S7-Programm enthält die Objekte “Quellen” und “Bausteine” als Software-Behälter und die Symboltabelle

Im Behälter “Bausteine” befinden sich das Objekt für den OB 1 und “Systemdaten” mit den übersetzten Konfigurationsdaten.

3.5 Verbindungen projektieren in der Verbindungstabelle

In S5 werden Verbindungen mit dem COM NCM projektiert. Es gibt für jeden CP ein COM-Paket. In S7 werden alle Verbindungen in der Verbindungstabelle projektiert.

Übersicht

Das Projektieren von Verbindungen ist die Voraussetzung für den Einsatz von SFB-Kommunikationsfunktionen im Anwenderprogramm.

Eine Verbindung legt folgendes fest:

- die beteiligten Kommunikationspartner im S7-Projekt
- den Typ der Verbindung (z.B. S7-homogene-Verbindung, FDL-Verbindung)
- spezielle Eigenschaften wie aktiver/passiver Verbindungsaufbau oder ob Betriebszustandsmeldungen gesendet werden sollen.

Bei der Verbindungsprojektierung wird pro Verbindung eine eindeutige lokale Kennung, die sogenannte lokale ID, vergeben. Nur diese lokale ID wird bei der Parametrierung der Kommunikationsfunktion benötigt.

Für jede CPU, die Endpunkt einer Verbindung sein kann, existiert eine eigene Verbindungstabelle.

Besonderheit

Sind beide Kommunikationspartner S7-400-Stationen, so wird automatisch für beide Endpunkte der Verbindung je eine lokale ID vergeben. Bei Verbindungen zu einer S7-300-Station wird nur eine lokale ID auf der S7-400-Station generiert.

Laden der Konfigurationsdaten

Die lokalen Konfigurationsdaten der Verbindungsendpunkte auf einer S7-Station müssen explizit in jede Zielstation geladen werden.

Eine (leere) Verbindungstabelle (Objekt ”Verbindungen”) wird automatisch zu jeder CPU angelegt. Die Verbindungstabelle wird zur Definition von Kommunikationsverbindungen zwischen CPUs in einem Netz verwendet. Nach dem Öffnen wird ein Fenster aufgeblendet und darin eine Tabelle zur Definition von Verbindungen zwischen programmierbaren Baugruppen angezeigt (Definieren von Verbindungen siehe Benutzerhandbuch **/231/**).

Beispiel: Verbindung zu einer S5

Das Beispiel zeigt Ihnen, wie Sie eine Verbindung zu einer SIMATIC S5-Station projektieren. Sie haben bereits eine SIMATIC 400-Station in Ihrem Projekt eingefügt.

- Fügen Sie eine SIMATIC S5-Station in Ihr Projekt ein und stellen Sie die Eigenschaften der Station ein.
- Öffnen Sie die Verbindungstabelle der S7-Station und fügen Sie eine Verbindung mit dem Menübefehl **Einfügen > Verbindungen** ein. Es erscheint ein Dialogfenster, in dem Sie nun den Kommunikationspartner -die SIMATIC S5-Station- und die Verbindungsart eintragen können.
- Haben Sie die Informationen eingetragen, erscheint die Verbindung in der Verbindungstabelle. Die Eigenschaften für die Verbindung müssen in S5 in den entsprechenden COM NCM für die S5-Station eingetragen werden.

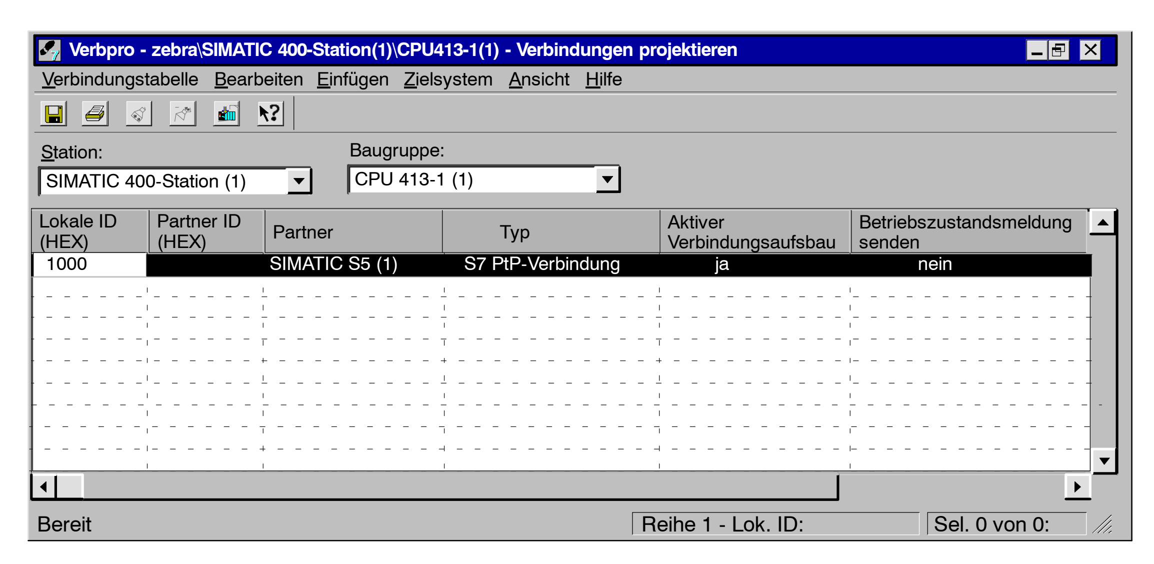


Bild 3-3 Verbindungstabelle

3.6 Programm einfügen und bearbeiten

Die Vorgehensweise, die in diesem Abschnitt beschrieben ist, bezieht sich auf die Neuerstellung eines Programms.

3.6.1 Prinzipielle Vorgehensweise bei der Software-Erstellung

Übersicht

Die Software für CPUs wird in Programm-Behältern abgelegt. Für SIMATIC S7-Baugruppen heißt ein solches Objekt ”S7-Programm”.

Das Bild zeigt z. B. ein S7-Programm in einer CPU einer SIMATIC 300-Station.

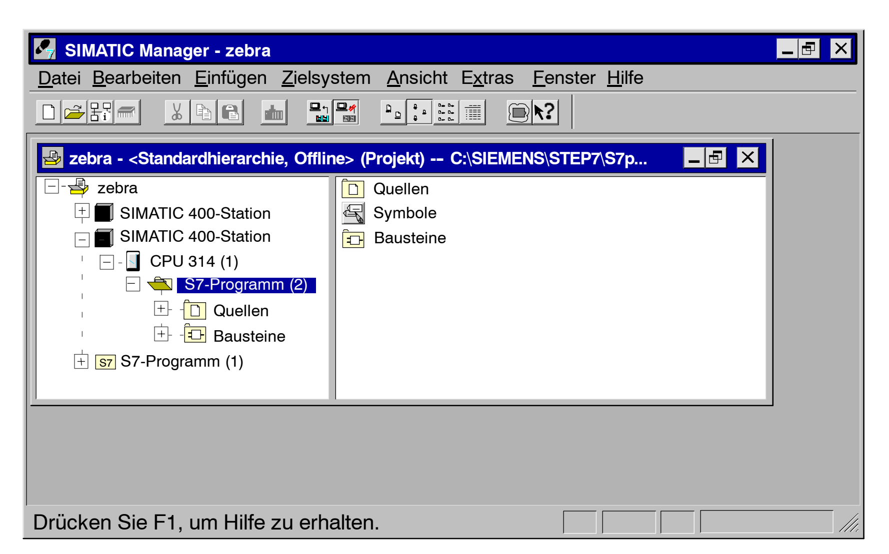


Bild 3-4 Geöffnetes S7-Programm im SIMATIC Manager

Vorgehensweise

Gehen Sie folgendermaßen vor, um die Software für Ihr Projekt zu erstellen:

- Öffnen Sie das S7-Programm.
- Öffnen Sie das Objekt ”Symbole” im S7-Programm und legen Sie die Symbole fest. (Dieser Schritt ist auch zu einem späteren Zeitpunkt durchführbar.) Weitere Information dazu finden Sie im Abschnitt 3.13.2.
- Öffnen Sie den Behälter ”Bausteine”, wenn Sie Bausteine erstellen wollen, oder den Behälter ”Quellen”, wenn Sie ein Quellprogramm erstellen wollen.
- Fügen Sie einen Baustein oder eine Quelle ein (Einzelheiten im Abschnitt 3.6.2). Die Menübefehle dazu lauten:
  - **Einfügen > S7-Software > Baustein...**, bzw.
  - **Einfügen > S7-Software > Quelle**
- Öffnen Sie den Baustein bzw. die Quelle und geben Sie ein Programm ein. Informationen dazu finden Sie in den Programmierhandbüchern /**232**/-/**236**/.

Je nach Anwendung sind nicht alle Punkte bei der Bearbeitung erforderlich.

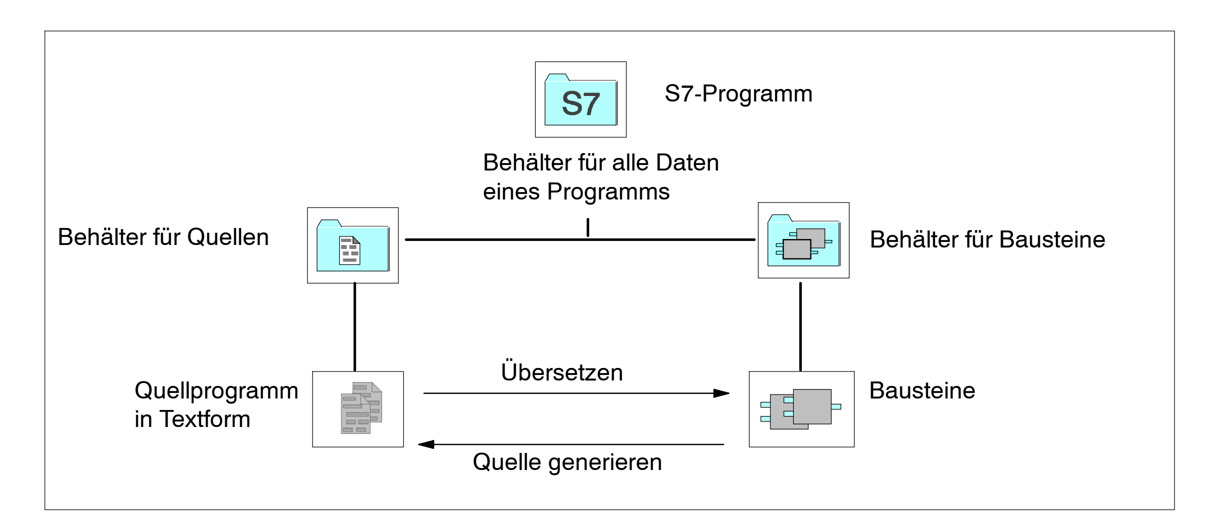


Bild 3-5 Wesentliche Objekte eines Projekts bei STEP 7 und ihre hierarchische Struktur

3.6.2 Einfügen von Komponenten für die Software-Erstellung in S7/M7-Programme

Bereits angelegte Komponenten

Zu jeder programmierbaren Baugruppe wird automatisch ein S7/M7-Programm als Behälter für die Software angelegt.

Das S7-Programm enthält bereits:

- Symboltabelle (Objekt ”Symbole”),
- einen Behälter ”Bausteine” für Bausteine mit erstem Baustein OB1,
- einen Behälter ”Quellen” für Quellprogramme.

S7-Bausteine erstellen

Sie wollen AWL-, KOP- oder FUP-Programme erstellen. Dazu markieren Sie das bereits angelegte Objekt ”Bausteine” und klicken dann auf den Menübefehl **Einfügen > S7-Software >Baustein**. In einem Folgemenü können Sie den Bausteintyp auswählen (Datenbaustein, Datentyp (UDT), Funktion, Funktionsbaustein, Organisationsbaustein, Variablentabelle (VAT)).

Nach dem Öffnen des (leeren) Bausteins können Sie das AWL-, KOP- oder FUP-Programm eingeben. Weitere Information dazu finden Sie in den Handbüchern zu AWL /**232**/, KOP /**233**/ und FUP /**236**/.

Das Objekt Systemdaten (SDB), das Sie eventuell in Anwenderprogrammen vorfinden, wird vom System erzeugt. Sie können das Objekt zwar öffnen, den Inhalt aus Konsistenzgründen aber nicht ändern. Es dient dazu, nach dem Laden eines Programms Konfigurationsänderungen vorzunehmen und diese auf das Zielsystem nachladen zu können.

Bausteine aus Standardbibliotheken benutzen

Für die Erstellung von Anwenderprogrammen können Sie auch Bausteine aus den Standardbibliotheken benutzen, die im Lieferumfang enthalten sind. Auf Bibliotheken greifen Sie über den Menübefehl **Datei > Öffnen** zu. Weitere Hinweise zum Benutzen von Standardbibliotheken sowie zum Anlegen eigener Bibliotheken finden Sie in der Online-Hilfe.

Quellen erstellen

Sie wollen eine Quelle in AWL erstellen. Dazu markieren Sie im S7-Programm das Objekt ”Quellen” bzw. ”Pläne” und klicken anschließend auf den Menübefehl **Einfügen > S7-Software > Quelle**. Wählen Sie im Folgemenü die zur Programmiersprache passende Quelle aus. Nach dem Öffnen der leeren Quelle können Sie das Programm eingeben.

Symboltabelle erstellen

Eine (leere) Symboltabelle (Objekt ”Symbole”) wird beim Erzeugen des S7-Programms automatisch angelegt. Nach dem Öffnen wird das Fenster ”Symbol Editor” aufgeblendet und darin die Symboltabelle angezeigt (siehe Abschnitt 3.13.2*, Symbolische Adressierung*).

Externe Quelle einfügen

Sie können Quelldateien mit beliebigen ASCII-Editoren erstellen und bearbeiten. Diese Dateien können Sie anschließend in ein Projekt importieren und in ablauffähige Bausteine übersetzen. Gehen Sie folgendermaßen vor:

- Selektieren Sie dazu den Behälter ”Quellen”, in den die Quelldatei importiert werden soll.
- Wählen Sie den Menübefehl **Einfügen > Externe Quelle**.
- Geben Sie im aufgeblendeten Dialogfeld die Quelldatei an.

Die bei der Übersetzung einer importierten Quelle entstehenden Bausteine werden im Behälter ”Bausteine” abgelegt.

3.7 Bausteine

3.7.1 Gegenüberstellung

Die folgende Tabelle stellt die Bausteine von STEP 5 und STEP 7 gegenüber. Die Tabelle gibt Antwort auf die Frage ”Welchen STEP 7-Baustein nehme ich für welchen STEP 5-Baustein?”

Keine starre Zuordnung

Die Tabelle ist nicht als starre 1-zu-1-Zuordnung zu verstehen, da die neue Bausteinwelt zusätzliche Möglichkeiten für die Programmierung offen hält. Es handelt sich um eine Empfehlung für den Einstieg in die Programmierung mit STEP 7.

Tabelle 3-2 Gegenüberstellung von STEP 5- und STEP 7-Bausteinen

STEP 5-Baustein	STEP 7-Baustein	Erläuterung
Organisationsbaustein (OB)	Organisationsbausteine (OB)	Schnittstelle zum Betriebssystem
Integrierte Sonder-OBs	Systemfunktionen (SFC) Systemfunktionsbausteine (SFB)	Systemfunktionen bei STEP 7 sind Ersatz für Sonder-Organisationsbausteine (STEP 5), die im Anwenderprogramm aufgerufen werden können.
Funktionsbaustein (FB, FX)	Funktion (FC)	Funktionen (FC) bei STEP 7 haben die gleichen Eigenschaften wie Funktionsbausteine bei STEP 5.
Programmbaustein (PB)	Funktionsbaustein (FB)	Programmbausteine entsprechen in STEP 7 den Funktionsbausteinen. Funktionsbausteine in STEP 7 haben völlig neue Eigenschaften verglichen mit den namensgleichen Bausteinen bei STEP 5. Sie eröffnen damit auch neue Möglichkeiten der Programmierung. Beachte: Programmbausteine werden bei der Konvertierung in Funktionen (FCs) umgesetzt.
Schrittbaustein (SB)	-	Schrittbausteine entfallen in STEP 7.
Datenbaustein (DB, DX)	Datenbaustein (DB)	In STEP 7 sind die Datenbausteine länger als bei STEP 5 (bei S7-300 bis 8 kByte, bei S7-400 bis 64 kByte)
Datenbaustein DX0, DB1 in ihrer besonderen Funktion	Systemdatenbausteine (SDB) (CPU-Parametrierung)	Die neuen Systemdatenbausteine enthalten alle Daten der Hardware-Konfiguration, also auch die CPU-Parametrierungen, die die Programmbearbeitung festlegen.
Kommentarbausteine DK, DKX, FK, FKX, PK	-	In STEP 7 gibt es keine Kommentarbausteine mehr. Der Kommentar ist in den entsprechenden Bausteinen der Offline-Datenhaltung enthalten.

3.7.2 Funktionen und Funktionsbausteine

Funktionen (FC)

Eine Funktion (FC) ist ein Codebaustein ”ohne Gedächtnis”. Die Ausgangsparameter enthalten nach der Bearbeitung der FC die berechneten Funktionswerte. Die weitere Verwendung und Speicherung der Aktualparameter nach dem Aufruf einer FC liegt in der Hand des Anwenders.

Verwechseln Sie Funktionen bitte nicht mit Funktionsbausteinen! Beides sind bei STEP 7 verschiedene Arten von Bausteinen.

Funktionsbausteine (FB)

Ein Funktionsbaustein (FB) ist ein Codebaustein ”mit Gedächtnis”. Als Gedächtnis dient dabei ein dem FB zugeordneter Instanz-Datenbaustein, in dem die Aktualparameter und statischen Daten des Funktionsbausteins gespeichert werden.

Anwendungsbereiche von Funktionsbausteinen sind z. B. Programmierung von Reglerstrukturen.

3.7.3 Datenbausteine

Datenbausteine speichern die Daten des Anwenderprogramms. Bei den Datenbausteinen wird zwischen Global-Datenbausteinen und Instanz-Datenbausteinen unterschieden:

- Global-Datenbausteine sind keinem Baustein fest zugeordnet (wie bei STEP 5).
- Instanz-Datenbausteine sind einem Funktionsbaustein zugeordnet und enthalten neben den Daten des FBs auch die Daten eventuell definierter Multiinstanzen.

Jeder Datenbaustein kann entweder ein Global-Datenbaustein oder ein Instanz-Datenbaustein sein.

3.7.4 Systembausteine

Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB)

Nicht jede Funktion müssen Sie selbst programmieren. Sie können auf vorgefertigte Bausteine zurückgreifen, die im Betriebssystem der Zentralbaugruppen vorhanden sind, z. B. für die Programmierung von Kommunikationsfunktionen. Im einzelnen sind das folgende Bausteine:

- **Systemfunktionen (SFC)**, mit Eigenschaften wie Funktionen (FC),
- **Systemfunktionsbausteine (SFB)**, mit Eigenschaften wie Funktionsbausteine (FB).

Systemdatenbausteine (SDB)

Bisher war die Rede von Bausteinen, die Programm oder Daten des Anwenderprogramms enthalten. Neben diesen Bausteinen gibt es Bausteine, die Einstellungen wie z. B. Parameter von Baugruppen oder Adressen enthalten. Sie heißen **Systemdatenbausteine (SDB)**. Systemdatenbausteine werden von speziellen STEP 7-Tools erzeugt, z.B. bei der Eingabe der Hardware-Konfiguration oder bei der Erstellung von Verbindungstabellen.

3.7.5 Organisationsbausteine

Organisationsbausteine (OBs) stellen die Schnittstelle zwischen Betriebssystem und Anwenderprogramm dar. Verschiedene Organisationsbausteine übernehmen dabei ganz bestimmte Aufgaben.

Einteilung der Organisationsbausteine

Das AWL-Anwenderprogramm für Ihre S7-CPU stellen Sie aus den Organisationsbausteinen (OBs) zusammen, die Sie für Ihre Automatisierungslösung benötigen.

Tabelle 3-3 Gegenüberstellung der OBs in S5 und S7

Funktion		S5	S7
Hauptprogramm	Freier Zyklus	OB 1	OB 1
Alarme	Verzögerungsalarm	OB 6	OB 20 bis OB 23
	Uhrzeitalarm	OB 9	OB 10 bis OB 17
	Hardware-Interrupts	OB 2 bis OB 5	OB 40 bis OB 47
	Prozeßalarme	OB 2 bis OB 9 (EB 0)	Werden ersetzt durch Interrupts
	Weckalarme	OB 10 bis OB 18	OB 30 bis OB 38
	Mehrprozessor-Alarm	-	OB 60
Anlauf	Manueller Neustart	OB 21 (AG 115U) OB 20 (ab AG 135U)	OB 100
	Manueller Wiederanlauf	OB 21 (ab AG 135U)	OB 101
	Automatischer Wiederanlauf	OB 22	OB 101
Fehler	Fehler	OB 19 bis OB 35	OB 121, OB 122, OB 80 bis OB 87
Sonstiges	Bearbeitung im STOP-Zustand	OB 39	entfällt
	Hintergrundbearbeitung	-	OB 90

Fehlerbehandlung

Fehler-OBs

Fehler-OBs werden aufgerufen, wenn im Programmablauf ein Fehler auftritt. Mit ihrer Hilfe können Sie Fehlerreaktionen programmieren. Existiert zu einem Fehlertyp kein Fehler-OB, geht die CPU in den Stop-Zustand.

Tabelle 3-4 Gegenüberstellung der Fehler-OBs in S5 und S7

Funktion	S5	S7
Aufruf eines nicht geladenen Bausteins	OB 19	OB 121
Quittungsverzug bei Direktzugriff auf Peripheriebaugruppen	OB 23	OB 122
Quittungsverzug beim Aktualisieren des Prozeßabbildes und der Koppelmerker	OB 24	OB 122
Adressierfehler	OB 25	OB 122
Zykluszeitüberschreitung	OB 26	OB 80
Substitutionsfehler	OB 27	entfällt
Stop durch Bedienung	OB 28 (AG 135U)	entfällt
Quittungsverzug beim Eingangsbyte EB 0	OB 28 (AG 155U)	OB 85
Nicht zulässiger Operationscode	OB 29 (AG 135U)	STOPP
Quittungsverzug beim Direktzugriff auf Peripherie im erweiterten Adressierbereich	OB 29 (AG 155U)	OB 122
Nicht zulässiger Parameter	OB 30 (AG 135U)	entfällt
Parityfehler oder Quittungsverzug beim Zugriff auf den Anwenderspeicher	OB 30 (AG 155U)	OB 122
Sonderfunktions-Sammelfehler	OB 31	entfällt
Transferfehler bei Datenbaustein	OB 32	OB 121
Weckfehler bei zeitgesteuerter Bearbeitung	OB 33	OB 80
Reglerfehler	OB 34 (AG 135U)	entfällt
Fehler beim Erzeugen eines Datenbausteins	OB 34 (AG 155U)	Rückmeldung des SFC
Schnittstellenfehler	OB 35	OB 84

Fehlerbehebung in S5 und S7

Bereichsüberschreitung

Wie in S5 können Sie auch in S7 die Meldung einer Bereichsüberschreitung über Statusbits OV und OS auswerten. Das Verhalten weist geringfügige Unterschiede auf.

Im AWL-Handbuch /**232**/ können Sie das Verhalten der Statusbits in Zusammenhang mit den Anweisungen nachlesen.

Integrierte Sonderfunktionen

Bei S5-CPUs besteht die Schnittstelle zwischen Anwenderprogramm und dem Systemprogramm aus Zugriffen auf den BS-Bereich und über Sonder-OBs.

Bei S7-CPUs gibt es dafür zusätzlich zu den Organisationsbausteinen die neuen Bausteinarten ”Systemfunktionen” und ”Systemfunktionsbausteine”.

Systemfunktionen/ Systemfunktionsbausteine

Systemfunktionen (SFCs) und Systemfunktionsbausteine (SFBs) sind im Betriebssystem der CPU integrierte Bausteine, die bei Bedarf im STEP 7-Anwenderprogramm aufgerufen werden können. Tritt bei der Bearbeitung einer Systemfunktion SFC ein Fehler auf, kann dieser mit Hilfe des Rückgabewertes RET_VAL im Anwenderprogramm ausgewertet werden.

Tabelle 3-5 Sonderfunktionen in S5 und S7

Funktion	S5-Baustein	Ersatz in S7
Zykluszeittriggerung	OB 31	SFC 43 RE_TRIGR
Batterieausfall	OB34	OB 81 (Fehlerreaktion vom Anwender programmierbar)
Zugriff auf Anzeigenbyte	OB 110	STEP 7-Befehl: L STW/T STW
AKKU 1 - 4 löschen	OB 111	STEP 7-Befehlsfolge: L 0; PUSH; PUSH; PUSH
AKKU Roll Up	OB 112	Mit abweichender Funktion: STEP 7-Befehl: PUSH
AKKU Roll Down	OB 113	Mit abweichender Funktion: STEP 7-Befehl: POP
Alarme gemeinsam sperren ein-/ausschalten	OB 120	SFC 41 DIS_AIRT SFC 42 EN_AIRT
Weckalarme einzeln sperren ein-/ausschalten	OB 121	SFC 39 DIS_IRT SFC 40 EN_IRT
Alarme gemeinsam verzögern ein-/ausschalten	OB 122	SFC 41 DIS_AIRT SFC 42 EN_AIRT
Weckalarme einzeln verzögern ein-/ausschalten	OB 123	SFC 39 DIS_IRT SFC 40 EN_IRT
CPU-Uhrzeit stellen/lesen *(Fortsetzung auf nächster Seite)*	OB 150	SFC 0 SET_CLK SFC 1 READ_CLK

Tabelle 3-5 Sonderfunktionen in S5 und S7, Fortsetzung

Funktion	S5-Baustein	Ersatz in S7
Uhrzeitgesteuerte Weckzeit stellen/lesen	OB 151	SFC 28 SET_TINT SFC 30 ACT_TINT SFC 31 QRY_TINT
Zyklusstatistik	OB 152	Lokaldaten im OB 1
Zählschleife	OB 160 - 163 (AG 135U)	STEP 7-Befehl: LOOP
Variable Zeitschleife	OB 160 (AG 115U)	SFC 47 WAIT
Bausteinstack lesen	OB 170	entfällt
Variabler Datenbaustein-Zugriff	OB 180	entfällt
Datenbaustein testen	OB 181	SFC 24 TEST_DB
Datenbereich kopieren	OB 182	SFC 20 BLKMOV
Merker in Datenbausteine übertragen	OB 190, 192	SFC 20 BLKMOV
Datenblöcke in Merkerbereiche übertragen	OB 191, 193	SFC 20 BLKMOV
Funktionen zur Mehrprozessor-kommunikation	OB 200 - 205	entfällt
Kachelzugriffe	OB 216 - 218	Keine Kacheladressierung bei S7
Vorzeichenerweiterung	OB 220	S7-Befehl: ITD
Zyklusüberwachungszeit einstellen	OB 221	Parametrierung mit S7
Zyklusüberwachungszeit neu starten	OB 222	SFC 43 RE_TRIGR
Anlaufarten vergleichen	OB 223	Mehrprozessoranlauf nur bei gleicher Anlaufart
Koppelmerker blockweise übertragen	OB 224	entfällt
Wort aus dem Systemprogramm lesen	OB 226	entfällt
Quersumme des Systemprogramms lesen	OB 227	entfällt
Statusinformation einer Programmbearbeitungsebene lesen	OB 228	SFC 51 RDSYSST SFC 6 RD_SINFO
Funktionen für Hantierungsbausteine	OB 230 - 237	Kommunikation mit SFBs
Schieberegister initialisieren	OB 240	entfällt
Schieberegister bearbeiten	OB 241	entfällt
Schieberegister löschen *(Fortsetzung auf nächster Seite)*	OB 242	entfällt

Tabelle 3-5 Sonderfunktionen in S5 und S7, Fortsetzung

Funktion	S5-Baustein	Ersatz in S7
Regelung: PID-Algorithmus initialisieren Regelung: PID-Algorithmus bearbeiten	OB 250 OB 251	Regelungs-FBs: FB 41 - FB 43 oder SFB 41 - SFB 43
Datenbausteine (DB/DX) ins DB-RAM übertragen	OB 254, 255	entfällt

3.7.6 Bausteinabbildung bei der Konvertierung

Baustein-zuordnung

Die Bausteinstruktur wurde bei S7 geändert. Das Bild zeigt vereinfacht die Zuordnung von Bausteinen bei STEP 5 und bei STEP 7, wie sie bei der Konvertierung erfolgt.

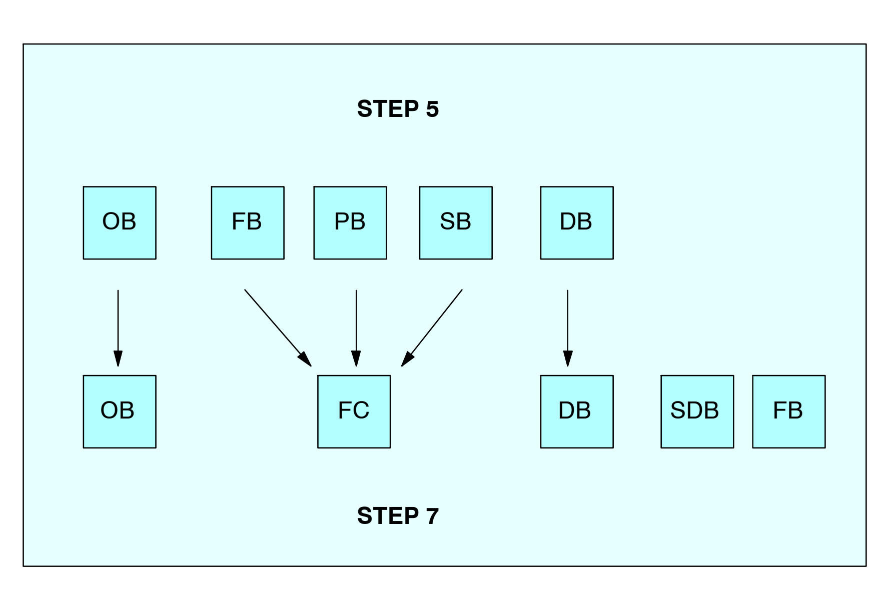


Bild 3-6 Bausteine vergleichbarer Funktion bei STEP 5 und bei STEP 7

Tabelle 3-6 auf Seite 3-26 zeigt Ihnen, wie Bausteinaufrufe bei der Konvertierung umgesetzt werden.

Tabelle 3-6 Bausteintypen in S5 und S7

S5			S7	
OB	feste Nummern	Anwenderprogramm	Entsprechender S7-OB	feste Nummern
OB	feste Nummern	Sonderfunktionen	Nicht konvertierbar; muß bei S7 neu programmiert werden.	
PB	0 bis 255	Anwenderprogramm	FC-Bausteine ohne Parameter	Nummer wird vorgeschlagen.
FB/FX	0 bis 255	Anwenderprogramm	FC-Bausteine mit Parameter, deren Name erhalten bleibt	Nummer wird vorgeschlagen.
FB	feste Nummern	Integrierte Funktionsbausteine	Ladbare FC, die in der Bibliothek FBLib1 enthalten sind und vor dem Übersetzen in die konvertierte Datei geladen werden müssen	feste Nummern
FB/FX	feste Namen	Standard-Funktionsbausteine	Ladbare FC, die in der Bibliothek FBLib1 enthalten sind und vor dem Übersetzen in die konvertierte Datei geladen werden müssen	feste Nummern
SB	0 bis 255	Anwenderprogramm	FC-Bausteine ohne Parameter Ablaufketten sind nicht konvertierbar und müssen in GRAPH für S7 erstellt werden.)	Nummer wird vorgeschlagen.
DB	2 bis 255	Anwenderdaten	Globale Datenbausteine DB	Nummer wird von S5 übernommen.
DX	1 bis 255	Anwenderdaten	Globale Datenbausteine DB	Nummer ab 256 wird vorgeschlagen.
DB 1/ DX 0		Datenbausteine mit Systemeinstellungen	Enthalten die Bausteine CPU-spezifische Einträge, müssen die entsprechenden Einstellungen beim Parametrieren mit STEP 7 vorgenommen werden. Der konvertierte Bausteininhalt von DB 1 und DX 0 ist irrelevant und kann gelöscht werden.	

3.8 Systemeinstellungen

DB 1 und DX 0 umsetzen

Die Tabellen zeigen, wie die Funktionen der Parameter in DB 1 und DX 0 (Systemeinstellungen) in S7 realisiert sind:

Tabelle 3-7 Umsetzung der Systemeinstellungen aus DB 1

S5-Parameterblock	In S7 wie folgt realisiert
Anlaufverzögerung	Aufruf des SFC 47 WAIT
Koppelmerker	Wird eingestellt über Globaldaten-Kommunikation, Aufruf von: SFC 60 GD_SND SFC 61 GD_RCV
Lage des Errorcodes	System legt Fehlermeldungen im Diagnosepuffer ab. Angabe ”Lage des Errorcodes” entfällt.
Nummer integrierter FBs ersetzen	Entfällt
Onboard Analogeingänge	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Onboard Interrupt	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Onboard Zähler	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Prioritäten von OBs ändern	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Prozeßabbild ausgeben/sperren	Aufruf des SFC 27 UPDAT_PO
Prozeßabbild einlesen/sperren	Aufruf des SFC 26 UPDAT_PI
Remanente Merker	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Remanente Timer	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Remanente Zähler	Werden eingestellt in HWKonfig über CPU-Eigenschaften
SINEC L1	Wird ersetzt durch MPI-Bus (Globaldaten-Kommunikation)
SINEC L2	Einstellung mit HWKonfig
Softwareschutz	in Vorbereitung
Uhrzeit-Parameter	Werden eingestellt in HWKonfig über CPU-Eigenschaften oder über Aufruf des SFC 28 SET_TINT
Weckalarm-OBs parametrieren	Werden eingestellt in HWKonfig über CPU-Eigenschaften
Zykluszeit-Überwachung	Wird eingestellt in HWKonfig über CPU-Eigenschaften

Tabelle 3-8 Umsetzung der Systemeinstellungen aus DX 0

S5-Parameterblock	In S7 wie folgt realisiert
Adressierfehlerüberwachung	Aufruf von OB 121
Aktualisierung der Koppelmerker	Globaldaten-Kommunikation
Anlaufart nach Netzein	Wird eingestellt in HWKonfig über CPU-Eigenschaften
Anlauf-Synchronisation im Mehrprozessorbetrieb	Wird eingestellt in HWKonfig über CPU-Eigenschaften
Anzahl der Zeitzellen	CPU-spezifischer fester Wert (bei S7-300) oder einstellbar in HWKonfig über CPU-Eigenschaften (bei S7-400)
Fehlerbehandlung	Aufruf von: SFC 36 MSK_FLT SFC 37 DMSK_FLT
Gleitpunktarithmetik	vorhanden
Prozeßalarm-Triggerung	Wird eingestellt in HWKonfig über CPU-Eigenschaften
Weckalarm-Bearbeitungsmodus	Aufruf der SFC 28 SET_TINT
Zykluszeitüberwachung	Wird eingestellt in HWKonfig über CPU-Eigenschaften

3.9 Standardfunktionen

Bei der Konvertierung werden in S5 vorhandene Standardfunktionen automatisch durch konvertierte Funktionen mit der gleichen Funktionalität ersetzt. Diese Funktionen können in S7 meist durch einfache Befehlssequenzen ersetzt werden, wodurch Speicherplatz und Zykluszeit gespart werden können.

Die Standardfunktionen sind in der S7-Bibliothek "StdLib30" in dem Programmbehälter FBLib1 enthalten.

Hinweise zum Arbeiten mit Bibliotheken finden Sie in der Online-Hilfe.

3.9.1 Gleitpunktarithmetik

STEP 5	STEP 7		STEP 5	STEP 7	
FB-Name	**Nummer**	**Name**	**FB-Name**	**Nummer**	**Name**
GP:FPGP	FC 61	GP_FPGP	GP:MUL	FC 65	GP_MUL
GP:GPFP	FC 62	GP_GPFP	GP:DIV	FC 66	GP_DIV
GP:ADD	FC 63	GP_ADD	GP:VGL	FC 67	GP_VGL
GP:SUB	FC 64	GP_SUB	RAD:GP	FC 68	RAD_GP

3.9.2 Signalfunktionen

STEP 5	STEP 7		STEP 5	STEP 7	
FB-Name	**Nummer**	**Name**	**FB-Name**	**Nummer**	**Name**
MLD:TG	FC 69	MLD_TG	MLD:EZ	FC 75	MLD_EZ
MELD:TGZ	FC 70	MELD_TGZ	MLD:ED	FC 76	MLD_ED
MLD:EZW	FC 71	MLD_EZW	MLD:EZWK	FC 77	MLD_EZWK
MLD:EDW	FC 72	MLD_EDW	MLD:EDWK	FC 78	MLD_EDWK
MLD:SAMW	FC 73	MLD_SAMW	MLD:EZK	FC 79	MLD_EZK
MLD:SAM	FC 74	MLD_SAM	MLD:EDK	FC 80	MLD_EDK

3.9.3 Integrierte Funktionen

STEP 5	STEP 7	
FB-Name	**Nummer**	**Name**
COD:B4	FC 81	COD_B4
COD:16	FC 82	COD_16
MUL:16	FC 83	MUL_16
DIV:16	FC 84	DIV_16

3.9.4 Grundfunktionen

STEP 5	STEP 7		STEP 5	STEP 7	
FB-Name	Nummer	Name	FB-Name	Nummer	Name
ADD:32	FC 85	ADD_32	REG:LIFO	FC 93	REG_LIFO
SUB:32	FC 86	SUB_32	DB:COPY	FC 94	DB_COPY
MUL:32	FC 87	MUL_32	DB:COPY	FC 95	DB_COPY
DIV:32	FC 88	DIV_32	RETTEN	FC 96	RETTEN
RAD:16	FC 89	RAD_16	LADEN	FC 97	LADEN
REG:SCHB	FC 90	REG_SCHB	COD:B8	FC 98	COD_B8
REG:SCHW	FC 91	REG_SCHW	COD:32	FC 99	COD_32
REG:FIFO	FC 92	REG_FIFO			

3.9.5 Analogfunktionen

STEP 5	STEP 7		STEP 5	STEP 7	
FB-Name	Nummer	Name	FB-Name	Nummer	Name
AE:460	FC 100	AE_460_1	AE:466	FC 106	AE_466_1
AE:460	FC 101	AE_460_2	AE:466	FC 107	AE_466_2
AE:463	FC 102	AE_463_1	RLG:AA	FC 108	RLG_AA1
AE:463	FC 103	AE_463_2	RLG:AA	FC 109	RLG_AA2
AE:464	FC 104	AE_464_1	PER:ET	FC 110	PER_ET1
AE:464	FC 105	AE_464_2	PER:ET	FC 111	PER_ET2

3.9.6 Mathematische Funktionen

STEP 5	STEP 7		STEP 5	STEP 7	
FB-Name	Nummer	Name	FB-Name	Nummer	Name
SINUS	FC 112	SINUS	ARCCOT	FC 119	ARCCOT
COSINUS	FC 113	COSINUS	LN X	FC 120	LN_X
TANGENS	FC 114	TANGENS	LG X	FC 121	LG_X
COTANG	FC 115	COTANG	B LOG X	FC 122	B_LOG_X
ARCSIN	FC 116	ARCSIN	E^X	FC 123	E_H_N
ARCCOS	FC 117	ARCCOS	ZEHN^X	FC 124	ZEHN_H_N
ARCTAN	FC 118	ARCTAN	A2^A1	FC 125	A2_H_A1

3.10 Datentypen

STEP 7 verwendet neue Datenformate. Die Tabelle zeigt die unterschiedlichen Datentypen in S5 und S7.

Tabelle 3-9 Datentypen in S5 und S7

Datentypen in S5	Datentypen in S7	Art
BOOL, BYTE, WORD, DWORD, 16-bit-Festpunkt, 32-bit-Festpunkt, Gleitpunkt, Zeitwert, - (ASCII-Zeichen)	BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE; TIME_OF_DAY, CHAR	Elementare Datentypen
-	DATE_AND_TIME, STRING, ARRAY, STRUCT	Zusammengesetzte Datentypen
Zeiten, Zähler, Bausteine - -	TIMER, COUNTER, BLOCK_FC, BLOCK_FB, BLOCK_DB, BLOCK_SDB, POINTER, ANY	Parametertypen

Tabelle 3-10 Unterschiedliche Konstantenformate in S5 und in S7

Formate in S5	**Beispiel**	**Formate in S7**	**Beispiel**
KB	L KB 10	3#16#	L B#16# A
KF	L KF 10	-	L 10
KH	L KH FFFF	W#16#	L W#16# FFFF
KM	L KM 1111111111111111	2#	L 2# 11111111_11111111
KY	L KY 10,12	B#	L B# (10,12)
KT	L KT 10.0	S5TIME# (S5T#)	L S5TIME# 100ms
KZ	L KZ 30	C#	L C#30
DH	L DH FFFF FFFF	DW#16#	L DW#16# FFFF_FFFF
KC	L KC WW	’ xx ’	L ’ WW ’
KG	L KG +234 +09	REAL	L +2.34 E+08
Darstellung: S5-Format ← Exponent → ← Mantisse →		**Darstellung: Single Format nach ANSI/IEEE** V ← Exponent →← Mantisse →	
31 30 24 23 22 0 VE 2^6.. ... ... 2^0 VM 2^{-1}...... .....2^{-23}		31 30 23 22 0 V 2^7.. ... ... 2^0 2^{-1}.. .... 2^{-23}	
Exponent = Betrag des Exponenten VE = Vorzeichen des Exponenten VM = Vorzeichen der Mantisse		Exponent = tatsächl. Exponent + Bias* (+127) V = Vorzeichen der Mantisse	
Wertebereich: 1,5 x 10^{-39} bis 1,7 x 10^{38}		Wertebereich: ca. 1,18 x 10^{-38} bis 3,40 x 10^{+38}	

* Bias: Dies ist ein Offset, welcher den Exponenten in den positiven und negativen Bereich trennt. Der Wert 127 im Exponentenbereich entspricht absolut dem Wert 0.

Mehr Information zum Thema Datentypen finden Sie im AWL-Handbuch /**232**/.

3.11 Operandenbereiche

3.11.1 Übersicht

Tabelle 3-11 Operanden in S5 und S7

Operandenbereiche	Operanden in S5	entsprechende Operanden in S7	Anmerkung
Eingänge	E	E	
Ausgänge	A	A	
Peripherie	P, Q, G	PE → bei Ladebefehlen	Globalperipherie wird
		PA → bei Transferierbefehlen	nicht konvertiert
Merkerbereich	M	M	
	S	M	ab M 256.0 (Konverter)
	“Schmiermerker”	L	werden wie Merker konvertiert
Zeiten	T	T	
Zähler	Z	Z	
Datenbereich	D...	DB...	werden als Globaldatenoperanden konvertiert
Systemdaten	BS, BT, BA, BB	-	werden nicht
Kachelbereich	C	-	konvertiert

Anmerkung zu Datenoperanden

In S7 gibt es zwei Datenbausteinregister: das DB-Register, das überwiegend für globale Datenbausteine verwendet wird, und das DI-Register, das bevorzugt für Instanz-DBs eingesetzt wird. Deshalb gibt es zwei Typen von Datenoperanden. Die Operanden DBX, DBB, DBW, DBD sind Operanden von globalen Datenbausteinen, die Operanden DIX, DIB, DIW, DID sind Operanden von Instanz-DBs. Bei der Konvertierung werden für die Datenbaustein-Operanden D, DB, DW, DD Operanden von globalen Datenbausteinen eingesetzt.

Beachten Sie in diesem Zusammenhang auch die Konvertierung von Datenbausteinen (siehe Abschnitt 3.7.6)

Hinweis

Beachten Sie, daß bei S7 die Größe und die Nummernbereiche der Operandenbereiche und die Anzahl und Länge der Bausteine von der verwendeten CPU abhängen. Die Leistungsmerkmale der CPUs können Sie dem Abschnitt 2.2.1 entnehmen.

3.11.2 Neue Operanden in S7: Lokaldaten

Lokaldaten in STEP 7

Lokaldaten in STEP 7 sind die einem Codebaustein zugeordneten Daten, die in seinem Deklarationsteil bzw. seiner Variablendeklaration deklariert werden. Sie umfassen (bausteinabhängig): Bausteinparameter, statische Daten, temporäre Daten. Die Lokaldaten werden in der Regel symbolisch adressiert.

Bausteinparameter

Bausteinparameter von Funktionen (FC) werden wie die Bausteinparameter bei S5 behandelt: Die Bausteinparameter stellen Zeiger dar, die auf den entsprechenden Aktualparameter zeigen.

Bausteinparameter von Funktionsbausteinen (FB) werden wie die statischen Lokaldaten im Instanz-Datenbaustein abgelegt.

Statische Lokaldaten

Statische Lokaldaten können in jedem Funktionsbaustein verwendet werden. Sie werden im Deklarationsteil definiert und im Instanz-DB abgelegt.

Statische Lokaldaten behalten wie Datenoperanden in Global-Datenbausteinen ihren Wert so lange, bis sie durch das Programm überschrieben werden.

Gewöhnlich werden die statischen Lokaldaten nur im Funktionsbaustein bearbeitet. Da sie jedoch in einem Datenbaustein gespeichert sind, kann zu jedem Zeitpunkt im Anwenderprogramm auf die statischen Lokaldaten wie auf Variablen in einem Global-Datenbaustein zugegriffen werden.

Temporäre Lokaldaten

Schmiermerker in STEP 5

Für die Zwischenspeicherung von Daten innerhalb von Bausteinen wurden bei STEP 5 Merkerbereiche verwendet. Die Merker 200 ... 255 sind vereinbarungsgemäß als Zwischenspeicher reserviert. Die Verwaltung der Schmiermerker unterliegt vollständig dem Anwender.

Temporäre Lokaldaten in STEP 7

Temporäre Lokaldaten sind Speicherplätze für Daten, die nur während der Bearbeitung eines Bausteins gültig sind. Sobald der Baustein abgearbeitet ist, geben sie den benutzten Speicherplatz wieder frei. Jede Ablaufebene hat einen eigenen Lokaldatenstack. Versehentliches Überschreiben von Zwischenergebnissen durch Alarmprogramme ist damit ausgeschlossen.

Anwendung temporärer Lokaldaten in STEP 7

Temporäre Variablen werden bei STEP 7 für drei verschiedene Anwendungsbereiche genutzt:

- Als Zwischenspeicher für Daten des Anwenderprogramms.

  Diese Anwendung ist oben erläutert und gilt für die Bausteine ”Funktionen” (FC), ”Funktionsbausteine” (FB) und Organisationsbausteine (OB).

- Als Speicher zur Übergabe von Informationen des Betriebssystems an das Anwenderprogramm.

  Die vom Betriebssystem an das Anwenderprogramm gelieferten Informationen haben einen speziellen Namen: sie heißen ”Startinformation”. Startinformationen werden ausschließlich den Organisationsbausteinen (OB) als Schnittstelle zwischen Betriebssystem und Anwenderprogramm ”mitgegeben”.

- zur Parameterübergabe bei FCs.

Wo werden temporäre Lokaldaten vereinbart?

Temporäre Lokaldaten vereinbaren Sie innerhalb eines Bausteins. Wenn Sie einen Baustein neu erstellen, dann vereinbaren Sie zu Beginn Symbole für die temporären Variablen und verwenden Sie dann innerhalb des Bausteins. Pro Ablaufebene stehen in der S7-300 256 Bytes zur Verfügung. In der S7-400 stehen insgesamt bis zu 16 kByte zur Verfügung, die der Anwender bei der CPU-Parametrierung auf die Ablaufebenen aufteilen kann.

3.12 Operationen

Die folgende Tabelle gibt eine Übersicht über die Operationen. Dabei ist angegeben, welche Operationen konvertiert werden können. Sind die Operationen nicht konvertierbar, werden Umsetzungsmöglichkeiten angegeben.

Tabelle 3-12 Operationen in S5 und in S7

Operationstyp	Operationen in S5	Operationen in S7	Konvertierung	Empfohlene Umsetzung
Akkumulator-operationen	TAK, ENT, I, D, ADDBF, ADDKF, ADDDH	TAK, ENT, INC, DEC, +, Neu in S7: TAW, TAD, PUSH, POP, LEAVE	ja	-
Adreßregisteranweisungen / Registeranweisungen	MA1, MBR, ABR, MAS, MAB, MSB, MSA, MBA, MBS; TSG, LRB, LRW, LRD, TRB, TRW, TRD	Neu in S7: LAR1, LAR2, TAR1, TAR2, +AR1, +AR2, TAR	nein	Adreßregister (AR1, AR2) verwenden
Bitverknüpfungs-operationen	U, UN, O, ON, U(, O(,), O, S, R, RB, RD, = P, PN, SU, RU	U, UN, O, ON, U(, O(,), O, S, R, = SET; U, SET; UN, SET; S, SET; R Neu in S7: X, XN, X(, XN(,FP, FN, NOT, SET, CLR, SAVE	ja	-
Zeitoperationen	SI, SV, SE, SS/ SSV, SA/SAR, FR, SVZ	SI, SV, SE, SS, SA, FR, S T	ja	-
Zähloperationen	ZV/SSV, ZR/SAR, FR, SVZ	ZV, ZR, FR, S Z	ja	-

Tabelle 3-12 Operationen in S5 und in S7, Fortsetzung

Operationstyp	Operationen in S5	Operationen in S7	Konvertierung	Empfohlene Umsetzung
Lade- und Transferoperationen	L, LC, LW, LD, T L PB, L QB, L PW, L QW, T PB, T QB, T PW, T QW	L, LC, T L PEB, L PEW, T PAB, T PAW	ja	-
(Fortsetzung auf nächster Seite)	LB GB / GW / GD / CB / CW / CD, LW GW / GD / CW / CD, TB GB / GW / GD / CB / CW / CD, TW GW / GD / CW / CD		nein	durch Zugriff auf Peripheriebereich ersetzen
Festpunkt-arithmetik	+F, -F, XF, :F, +D, -D	+I, -I, *I, /I, +D, -D, *D, /D Neu in S7: MOD	ja	-
Gleitpunkt-arithmetik	+G, -G, XG, :G	+R, -R, *R, /R	ja	-
Vergleichs-operationen	!=F, ><F, >F, <F, >=F, <=F, !=D, ><D, D, <D, >=D, <=D, !=G, ><G, >G, <G, >=G, <=G	==I, <>I, >I, <I; >=I, <=I, ==D, <>D, >D, <D, >=D, <=D, ==R, <>R, >R, <R, >=R, <=R	ja	-
Umwandlungs-operationen	KEW, KZW, KZD DEF, DED, DUF, DUD, GFD, FDG	INVI, NEGI, NEGD, BTI, BTD, DTB, ITB, RND, DTR Neu in S7: ITD, RND+, RND-, TRUNC, INVD, NEGR	ja	-
Wort-verknüpfungs-operationen	UW, OW, XOW	UW, OW, XOW Neu in S7: UD, OD, XOD	ja	-
Schiebe- und Rotieroperationen	SLW, SLD, SRW, SRD, SVW, SVD, RLD, RRD	SLW, SLD, SRW, SRD, SSI, SSD, RLD, RRD Neu in S7: RLDA, RRDA	ja	-

Tabelle 3-12 Operationen in S5 und in S7, Fortsetzung

Operationstyp	Operationen in S5	Operationen in S7	Konver-tierung	Empfohlene Umsetzung
Datenbaustein-operationen	A, AX	AUF	ja	
	E, EX	SFC 22	nein	durch Aufruf SFC 22 CREATE_DB ersetzen
(Fortsetzung auf nächster Seite)		Neu in S7: TDB L DBLG, L DBNO, L DILG, L DINO		
Sprung-operationen	SPA, SPB, SPN, SPZ, SPP, SPM, SPO, SPS, SPR	SPA, SPB, SPN, SPZ, SPP, SPM, SPO, SPS Neu in S7: SPBN, SPBB, SPBNB, SPBI, SPBIN, SPMZ, SPPZ, SPU, LOOP, SPL	ja	-
Baustein-operationen	SPA, SPB, BA, BAB, BE, BEA, BEB	CALL, BE, BEA, BEB	ja	-
Befehlsausga-beoperationen / Master Control Relay-Operatio-nen	BAS, BAF	Neu in S7: MCRA, MCRD, MCR(,)MCR	nein	durch Aufruf SFC 26, SFC 27 ersetzen oder Master Control-Relay-Operationen
Stopbefehle	STP, STS, STW	SFC 46	nein	durch Aufruf SFC 46 STP ersetzen
Bearbeitungs-funktionen	B <Formalparame-ter>	-	nein	Aufruf von DB / Code-baustein muß neu pro-grammiert werden
	B MW, B DW	speicherindirekte Adressierung	ja	Empfehlung: Ersetzen durch registerindirekte Adressierung
	B BS	bereichsübergrei-fende registerindi-rekte Adressierung	nein	muß ersetzt werden durch indirekte Adres-sierung (siehe 3.13.4)
Absolute Spei-cheradressierung	LIR, TIR, LDI, TDI	-	nein	muß ersetzt werden durch indirekte Adres-sierung (siehe 3.13.4)
Blocktransfer	TNB, TNW, TXB, TXW	SFC 20	nein	durch Aufruf SFC 20 BLKMOV ersetzen

Tabelle 3-12 Operationen in S5 und in S7, Fortsetzung

Operationstyp	Operationen in S5	Operationen in S7	Konver-tierung	Empfohlene Umsetzung
Interruptbefehle	LIM, SIM, AFS, AFF, AS, AF	SFC 39 -42	nein	durch Aufruf SFC 39 - 42 ersetzen
Kachelbefehle	ACR, TSC, TSG	-	nein	Kachelzugriffe gibt es in S7 nicht mehr!
Mathematische Funktionen	-	ABS, COS, SIN, TAN , ACOS, ASIN, ATAN, EXP, LN	-	-
Nulloperationen	BLD xxx NOP 0, NOP 1	BLD xxx NOP 0, NOP 1	ja	-

3.13 Adressierung

3.13.1 Absolute Adressierung

Die absolute Adressierung ist in S5 und S7 identisch mit einer Ausnahme:

Daten in Datenbausteinen werden in S7 **byteweise** adressiert, d. h. Wort-Adressen bei S5 werden (durch Multiplikation mit 2) in Byte-Adressen gewandelt.

Die Tabelle zeigt die Zuordnung bei der Konvertierung (Datenbereichadressierung):

S5	S7
DL 0, 1, 2, 3, ...255	DBB 0, 2, 4, 6, ...510
DR 0, 1, 2, 3, ...255	DBB 1, 3, 5, 7, ...511
DW 0, 1, 2, 3, ...255	DBW 0, 2, 4, 6, ...510
DD 0, 1, 2, 3, ...254	DBD 0, 2, 4, 6, ...508
D x.y	DBX 2 x.y für 8 ≤ y ≤ 15 DBX (2 x+1).y für 0 ≤ y ≤ 7

3.13.2 Symbolische Adressierung

Die symbolische Adressierung in S5 wurde in S7 übernommen. Bei der Erstellung und dem Einsatz der Symbole gibt es neue Möglichkeiten. Bei der Programmierung ergeben sich keine Unterschiede.

Symbole bei STEP 5

Symbole haben Sie für STEP 5-Programme mit Hilfe des Symbolik-Editors vereinbart. Der Symbolik-Editor erzeugt eine Zuordnungsliste, die Ihnen erlaubt, anstelle von absoluten Adressen die in der Zuordnungsliste definierten Symbole zu verwenden.

Symbole bei STEP 7

Die Symbole können in STEP 7 bis zu 24 Zeichen lang sein.

Globale Symbole

Bei STEP 7 haben Sie ebenfalls einen Symbolik-Editor zur Verfügung. Die Zuordnungsliste heißt jetzt Symboltabelle. In ihr deklarieren Sie alle globalen Symbole (z. B. Eingänge, Ausgänge, Merker, Bausteine).

Wenn Sie Symbole mit dem Symbol-Editor zuweisen, dann gelten diese Symbole für ein CPU-Programm.

Lokale Symbole

Außer der Vereinbarung der Symbole mit dem Symbolik-Editor haben Sie bei STEP 7 die Möglichkeit, lokale Symbole für Datenoperanden und für den Lokaldatenbereich bei der Bausteinprogrammierung festzulegen.

Wenn Sie Symbole nicht mit dem Symbol-Editor zuweisen, sondern innerhalb eines Bausteins vereinbaren, dann ”gilt” dieses Symbol auch nur für den betreffenden Baustein. Wir sagen, das Symbol ist bausteinlokal.

Wann werden Symbole vereinbart ?

STEP 7 schreibt Ihnen nicht vor, zu welchem Zeitpunkt Symbole feststehen müssen. Sie haben folgende Möglichkeiten:

- vor Beginn der Programmierung

  (erforderlich bei inkrementeller Eingabe des Anwenderprogramms, d.h. mit Syntaxprüfung nach jeder Programmzeile),
- nach Erstellung des Anwenderprogramms, aber vor der Übersetzung

  (erforderlich bei quellorientierter Eingabe des Anwenderprogramms , d.h. das Programm wird als ASCII-Datei (=Quelle) erstellt).

Importieren einer Symboltabelle

In S7 haben Sie die Möglichkeit, die Symboltabelle mit einem Editor Ihrer Wahl zu erstellen und zu bearbeiten.

Sie können Tabellen, die Sie mit einem anderen Werkzeug erzeugt haben, in ihre Symboltabelle importieren und hier weiterbearbeiten. Die Import-Funktion läßt sich beispielsweise benutzen, um unter STEP5/ST erzeugte Zuordnungslisten nach Konvertierung in die Symboltabelle aufzunehmen.

Zur Auswahl stehen die Dateiformate *.SDF, *.ASC, *.DIF und *.SEQ.

Um eine Symboltabelle zu importieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie im Projektfenster das S7-Programm, in dem sich die Symboltabelle befindet.
2. Öffnen Sie die Symboltabelle durch Doppelklick auf den Behälter “Symbole”.
3. Wählen Sie im Fenster der Symboltabelle den Menübefehl **Tabelle > Importieren**. Ein Dialogfeld wird angezeigt.
4. Markieren Sie im Dialogfeld die Symboltabelle, die Sie importieren wollen und klicken Sie auf die Schaltfläche “Öffnen”.
5. Überpüfen Sie die Datensätze der Symboltabelle und nehmen Sie bei Bedarf Korrekturen vor.
6. Speichern und schließen Sie die Symboltabelle.

Hinweis

Eine Symboltabelle vom Dateiformat *.SEQ, die von S5 nach S7 konvertiert wurde, kann nicht mehr nach S5 importiert werden. Zum Austausch von Symboltabellen zwischen S5 und S7 wird das Dateiformat *.DIF empfohlen.

Weitere Information zum Thema Symboltabelle finden Sie im Benutzerhandbuch **/231/**.

3.13.3 Neu: Komplett-Adressierung von Datenoperanden

Komplett-Adressierung von Datenoperanden bedeutet, daß der Datenbaustein zusammen mit dem Datenoperanden angegeben wird. Das war in S5 nicht möglich.

Die Komplett-Adressierung kann entweder nur absolut oder nur symbolisch erfolgen. Eine Mischung absolut und symbolisch innerhalb einer Anweisung ist nicht möglich.

Beispiel

L DB100.DBW6
L DB_MOTOR.DREHZAHL

DB_MOTOR ist das Symbol für den Datenbaustein DB 100 und ist in der Symboltabelle definiert. DREHZAHL ist ein Datenoperand, der im Datenbaustein deklariert worden ist. Dadurch ist die symbolische Angabe des Datenoperanden (DB_MOTOR.DREHZAHL) ebenso eindeutig wie die absolute Angabe (DB100.DBW6).

Der komplettadressierte Datenzugriff ist nur in Verbindung mit dem Global-Datenbausteinregister (DB-Register) möglich. Der AWL-Editor setzt bei den komplettadressierten Datenoperaden zwei Anweisungen ab:

1. Aufschlagen des Datenbausteins über das DB-Register (z. B. AUF DB 100)
2. Zugriff auf den Datenoperanden (z. B. L DBW 6)

Mögliche Operationen mit komplett-adressierten Datenzugriff

Sie haben die Möglichkeit, für alle Operationen, die für den Datentyp des angesprochenen Datenoperanden zugelassen sind, den komplettadressierten Zugriff zu verwenden.

Komplettadressierte Datenoperanden können auch an Bausteinparametern angegeben werden. Dies wird dringend empfohlen, da beim Aufruf eines Bausteines eventuell der Datenbaustein gewechselt wird und Sie durch die Komplettadressierung sicherstellen, daß der richtige Datenoperand (aus dem richtigen Datenbaustein) übergeben wird.

Gefahren der "Teiladressierung"

Prinzipiell ist es möglich, auf Datenoperanden wie bei STEP 5 zuzugreifen ("Teiladressierung").

Beispiel:

L DBW 6
L DREHZAHL

Bei STEP 7 ist die Teiladressierung jedoch nicht unproblematisch, da STEP 7 die Register der S7-300/S7-400-CPU bei verschiedenen Operationen von sich aus ändert. In einigen Fällen wird die DB-Nummer im DB-Register überschrieben.

In den folgenden Situationen kann das DB-Register überschrieben werden. Hier ist besondere Vorsicht geboten:

- Das DB-Register wird bei einem komplettadressierten Datenzugriff überschrieben.
- Wenn ein FB aufgerufen wird, wird das DB-Register des aufrufenden Bausteins überschrieben.
- Nach einem Aufrufbefehl an eine FC, die einen Parameter mit zusammengesetztem Datentyp übergibt (z. B. STRING, DATE_AND_TIME, ARRAY, STRUCT oder UDT), wird der Inhalt des DB-Registers des aufrufenden Bausteins überschrieben.
- Nachdem Sie einer FC einen Aktualparameter zugeordnet haben, der in einem DB gespeichert ist (z. B. DB100.DBX0.1), öffnet STEP 7 den DB (DB 100), indem der Inhalt des DB-Registers überschrieben wird.
- Nachdem ein FB einen Durchgangsparameter mit zusammengesetztem Datentyp adressiert hat (z. B. STRING, DATE_AND_TIME, ARRAY, STRUCT oder UDT), verwendet STEP 7 das DB-Register zum Zugreifen auf Daten. Dadurch wird der Inhalt des DB-Registers überschrieben.
- Nachdem eine FC einen Parameter (Eingang, Ausgang oder Durchgang) mit zusammengesetztem Datentyp adressiert hat (z. B. STRING, DATE_AND_TIME, ARRAY, STRUCT oder UDT), verwendet STEP 7 das DB-Register zum Zugreifen auf Daten. Dadurch wird der Inhalt des DB-Registers überschrieben.

3.13.4 Indirekte Adressierung

Die indirekte Adressierung mit Hilfe der Bearbeitefunktion von S5 wird in S7 durch die neuen speicher- und registerindirekt adressierenden Befehle ersetzt.

Pointerformat bei STEP 5

In S5 belegt der Zeiger für die indizierte Bearbeitungsoperation ein Wort. Der Aufbau des Zeigers wird in Bild 3-7 dargestellt:

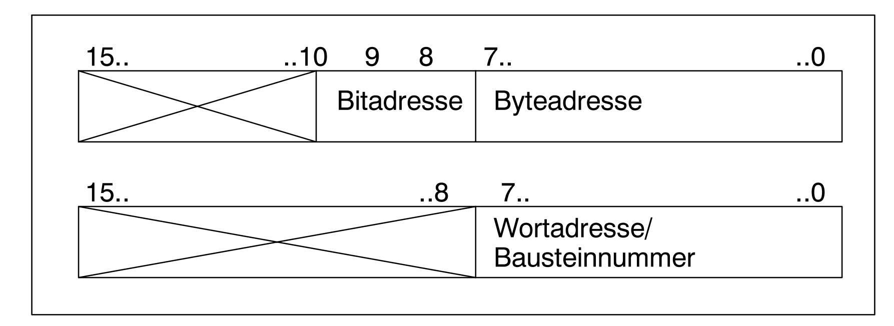


Bild 3-7 Aufbau Zeiger S5

Pointerformate bei STEP 7

Es gibt zwei mögliche Pointerformate in S7: Wort und Doppelwort.

15.. ..8	7.. ..0
n n n n n n n n	n n n n n n n n

Bits 0 bis 15 (nnnn nnnn nnnn nnnn): Nummer (Bereich 0 bis 65 535) einer Zeit (T), eines Zählers (Z), Datenbausteins (DB), einer Funktion (FC) oder eines Funktionsbausteins (FB)

Bild 3-8 Pointer im Wortformat für die speicherindirekte Adressierung

31.. ..24	23.. ..16	15.. ..8	7.. ..0
a 0 0 0 0 r r r	0 0 0 0 0 b b b	b b b b b b b b	b b b b b x x x

Bit 31 = 0 (a) gibt bereichs**interne** Adressierung an
Bit 31 = 1 (a) gibt bereichs**übergreifende** Adressierung an

Bit 24,25, 26 (rrr): Bereichskennung bei bereichsübergreifender Adressierung

Bits 3 bis 18 (bbbb bbbb bbbb bbbb): Nummer (Bereich 0 bis 65 535) des adressierten Byte

Bits 0 bis 2 (xxx): Nummer (Bereich 0 bis 7) des adressierten Bit

Bild 3-9 Pointer im Doppelwortformat für die speicherindirekte und die registerindirekte Adressierung

Speicherindirekte Adressierung

Die speicherindirekte Adressierung entspricht der indirekten Adressierung bei S5. Bei der speicherindirekten Adressierung gibt der Operand die Adresse des Werts an, den die Operation verarbeiten wird. Der Operand besteht aus den beiden folgenden Teilen:

- einem Operandenkennzeichen (z. B. ”EB” für ”Eingangsbyte”) und
- einem Wort, das die Nummer einer Zeit (T), eines Zählers (Z), Datenbausteins (DB), einer Funktion (FC) oder eines Funktionsbausteins (FB) enthält oder
- einem Doppelwort, das die genaue Adresse eines Werts innerhalb des Speicherbereichs enthält, den das Operandenkennzeichen angibt.

Der Operand gibt die Adresse des Werts oder der Nummer indirekt über den Pointer an. Dieses Wort oder Doppelwort kann sich in einem der folgenden Bereiche befinden:

- Merker (M)
- Datenbaustein (DB)
- Instanz-Datenbaustein (DI)
- Lokaldaten (L)

Vorteil der speicherindirekten Adressierung ist, daß Sie den Operanden der Anweisung während der Programmbearbeitung dynamisch modifizieren können.

Beispiele

Die folgenden beiden Beispiele zeigen, wie Sie mit einem Pointer im Wortformat arbeiten:

AWL S5	AWL S7	Erläuterung
L KB 5 T MW 2 B MW 2 L T 0	L +5 T MW 2 L T [MW 2]]	Lade den Wert 5 als Ganzzahl in AKKU 1. Transferiere den Inhalt von AKKU 1 ins Merkerwort MW2. Lade den Zeitwert der Zeit T 5.

Die folgenden beiden Beispiele zeigen, wie Sie mit einem Pointer im Doppelwortformat arbeiten:

AWL S5	AWL S7	Erläuterung
L KB 8 T MB 3 L KB 7 T MB 2 B MW 2 U E 0.0 B MW 2 = A 0.0	L P#8.7 T MD 2 U E [MD 2] = A [MD 2]	Lade 2#0000 0000 0000 0000 0000 0000 0100 0111 (Binärwert) in AKKU 1 (S7). Speichere die Adresse 8.7 im Merkerwort MW 2 (S5) / Merkerdoppelwort MD 2 (S7). Die Steuerung fragt den Eingang E 8.7 ab und weist seinen Signalzustand dem Ausgang A 8.7 zu.

AWL S5	AWL S7	Erläuterung
L KB 8 T MW 2 B MW 2 L EB 0 B MW 2 T MW 0	L P#8.0 T MD2 L EB [MD2] T MW [MD2]	Lade 2#0000 0000 0000 0000 0000 0000 0100 0000 (Binärwert) in AKKU 1 (S7). Speichere die Adresse 8 im Merkerwort MW 2 (S5) / Merkerdoppelwort MD 2 (S7). Die Steuerung lädt das Eingangsbyte EB 8 und transferiert den Inhalt ins Merkerwort MW 8.

Verwenden der korrekten Syntax

Wenn Sie mit einem speicherindirekten Operanden arbeiten, der im Speicherbereich des Datenbausteins gespeichert ist, müssen Sie zuerst den Datenbaustein öffnen, indem Sie die Operation Aufschlage Datenbaustein verwenden. Dann können Sie das Datenwort oder das Datendoppelwort als indirekten Operanden verwenden, wie im nachfolgenden Beispiel gezeigt:

```
AUF     DB 10
L       EB  [DBD 20]
```

Wenn Sie auf ein Byte, Wort oder Doppelwort zugreifen, vergewissern Sie sich zunächst, daß die Bitnummer des Pointers ”0” ist.

Registerindirekte Adressierung

Für die registerindirekte Adressierung werden in STEP 7 die Adreßregister AR 1 und AR 2 verwendet.

Bei der registerindirekten Adressierung gibt der Operand die Adresse des Werts an, den die Operation verarbeiten wird. Der Operand besteht aus den beiden folgenden Teilen:

- einem Operandenkennzeichen,
- einem Adreßregister und einem Pointer zur Angabe eines Versatzes, der zum Inhalt des Adreßregisters addiert wird, um die genaue Adresse, die die Operation verarbeiten soll, zu ermitteln. Der Pointer wird durch P#Byte.Bit angegeben.

Der Operand zeigt indirekt auf die Adresse des Werts, und zwar über das Adreßregister plus Versatz.

Eine Operation, die die bereichsinterne, registerindirekte Adressierung verwendet, verändert den Wert im Adreßregister nicht.

Weitere Informationen dazu finden Sie im AWL-Handbuch /**232**/.

Teil 2: Programmkonvertierung

4 Vorgehensweise

Die Programmierung von S7 in AWL ist weitgehend kompatibel zu S5-AWL, ebenso KOP in S7 zu S5-KOP und FUP in S7 zu S5-FUP. Wenn Sie also S5-Anwender sind und vorhandene Programme in S7 einsetzen möchten, wird Ihnen dieser Schritt erheblich erleichtert. Sie können auf Ihren erprobten S5-Programmen aufbauen und sie in S7-Programme umsetzen.

Wie gehen Sie vor ?

Die folgende Auflistung zeigt Ihnen, wie Sie bei der Umsetzung Ihres S5-Programms vorgehen können und in welchen Handbuchkapiteln Sie die erforderliche Informationen finden.

Die Liste ist exemplarisch und soll als Leitfaden dienen (einzelne Schritte können auch übersprungen werden).

4.1 S5-System analysieren

Bevor Sie Ihr S5-Programm konvertieren, sollten Sie klären, ob die Voraussetzungen für eine Umsetzung Ihres Programms gegeben sind.

Funktionalität der Baugruppen (siehe Kapitel 2)

Wie kann die Funktionalität Ihrer verwendeten S5-Baugruppen realisiert werden? Können Ihre S5-Baugruppen mit Hilfe von Adaptionskapseln oder Anschaltbaugruppen in S7 eingesetzt werden? Können Ihre S5-Baugruppen durch S7-Baugruppen ersetzt werden?

Systemeinstellungen (siehe Abschnitt 3.8)

Wie sind die benötigten Systemeinstellungen in S7 realisierbar?

Befehlsumfang (siehe Abschnitt 3.12)

Wie ist der von der S5-CPU verwendete Befehlsumfang mit Ihrer S7-CPU realisierbar?

Sind einzelne Befehle nicht konvertierbar, erhalten Sie eine Meldung mit den entsprechen Programmstellen und Sie müssen die Befehle selbst umprogrammieren.

Standardsoftware (siehe Abschnitt 3.9)

Sind die im zu konvertierenden Programm aufgerufenen S5-Standardfunktionsbausteine auch als Funktionen in S7 vorhanden?

Zum Lieferumfang der S7-Basissoftware gehören die bereits umgesetzten Standard-Softwarepakete für Gleitpunktarithmetik, Signalfunktionen, Integrierte Funktionen, Grundfunktionen und Mathematische Funktionen.

Sonderfunktionen (siehe Tabellen ab Seite 3-23)

Lassen sich die evtl. im S5-Programm verwendeten integrierten Sonderfunktionen ersetzen?

Welche Teile Ihres Programmes sollten in S7 neu programmiert werden?

In der Regel können nicht alle Teile eines Programmes konvertiert werden. Folgende Punkte helfen Ihnen bei der Entscheidung, ob Sie Ihr S5-Programm mit dem Konverter umsetzen lassen oder ob Sie Ihr S5-Programm in S7 neu erstellen.

- Programme, die nur digitale und binäre Verknüpfungen enthalten, benötigen keine Überarbeitung.
- Adressierung von Operanden mit Absolutadressen ist in S7 nicht möglich. Entsprechende Anweisungen (z. B. LIR, TIR, ...) werden nicht konvertiert. Wird in einem Programm häufig mit Absolutadressen gearbeitet, ist es sinnvoll, diese Programmteile, gegebenenfalls das gesamte Programm, neu zu schreiben.
- Die Bearbeitungsfunktionen (z. B. B MW, B DW) werden zwar zum Teil konvertiert, Sie können jedoch Speicherplatz sparen, wenn Sie die Funktionen in S7 neu programmieren. Die Funktionalität kann mit der indirekten Adressierung realisiert werden.
- Bei Bausteinaufrufen muß grundsätzlich die Belegung der Parameter überprüft und angepaßt werden, da die Aktualparameter bei der Konvertierung ohne Änderung übernommen werden.

4.2 S7-Projekt erstellen

STEP 7 stellt Ihnen zwei Möglichkeiten zur Verfügung, ein Projekt zu erstellen:

Erstellen eines Projektes mit dem STEP 7 Assistent

Mit Hilfe des STEP 7 Assistenten legen Sie in kürzester Zeit ein STEP 7 Projekt mit der CPU, die Sie einsetzen wollen, an. Anschließend können Sie mit der Programmierung beginnen.

Manuelles Anlegen eines Projektes

Weiterhin besteht die Möglichkeit, das Projekt manuell anzulegen. Die Vorgehensweise entnehmen Sie bitte Abschnitt 3.3.1.

4.3 Hardware konfigurieren

Zu diesem Zeitpunkt ist es sinnvoll, die Hardware zu konfigurieren, da in HWKonfig Daten ermittelt werden, die für die Vorbereitung der Konvertierung bereits verwendet werden können.

Wenn Sie sich noch nicht auf einen Hardware-Aufbau festlegen wollen, können Sie die Konfigurierung auch später durchführen.

Hardware festlegen

Mit Hilfe der Informationen im Kapitel 2 (Hardware) können Sie die benötigten S7- bzw. S5-Baugruppen für Ihren Aufbau auswählen und die Hardware-Konfigurationstabelle ausfüllen (siehe Abschnitt 3.4).

Adreßvergabe

Die Adreßvergabe für die Baugruppen erfolgt automatisch durch HWKonfig. Sie können die Adressen bei der Konvertierung bereits berücksichtigen.

Systemeinstellungen vornehmen

Bei der Parametrierung der CPU in HWKonfig können Sie Systemeinstellungen vornehmen, die in S5 im DB 1/DX 0 oder durch Systemdienste realisiert worden sind (siehe Abschnitt 3.4).

Remanenzverhalten bestimmen

Das Remanenzverhalten kann ebenfalls in den Parametrierdaten der CPU eingestellt werden. Das Remanenzverhalten ist jedoch abhängig von der Batteriepufferung (siehe Abschnitt 3.4).

Vorbereitung der Konvertierung 5

Überblick

Bereitstellen der benötigten Dateien (siehe Abschnitt 5.1)	• Programmdatei <Name>ST.S5D • Querverweisliste <Name>XR.INI • optional Zuordnungsliste <Name>Z0.SEQ
Operanden prüfen (siehe Abschnitt 5.2)	• Anzahl der Operanden • Anzahl der Bausteine
S5-Programm vorbereiten (siehe Abschnitt 5.3)	• Auswerten und Löschen der Datenbausteine DB 1 / DX 0 • Entfernen von Aufrufen von integrierten Bausteinen • Entfernen von Zugriffen auf Systemdatenbereich • Anpassen der Operandenbereiche • Zuweisen von Makros zu nichtkonvertierbaren Programmteilen • Löschen von Datenbausteine ohne Struktur bis auf 1 Datenwort
Makros erstellen (siehe Abschnitt 5.4)	• Befehlsmakros • OB-Makros

5.1 Bereitstellen der benötigten Dateien

Als Ausgangsbasis für die Umsetzung Ihres S5-Programms werden folgende Daten benötigt:

- die Programmdatei <Name> ST.S5D und
- die Querverweisliste <Name> XR.INI.

Die Querverweisliste wird zum Konvertieren benötigt, um die Programmstruktur und Aufrufhierarchie des S5-Programms zu erhalten.

Optionale Angabe

Wollen Sie in Ihrem Programm statt der absoluten Operanden symbolische Namen verwenden, benötigen Sie zur Erzeugung der konvertierten Zuordnungsliste zusätzlich

- die S5-Zuordnungsliste <Name> Z0.SEQ.

Vorgehensweise

Bereiten Sie die Konvertierung folgendermaßen vor:

1. Legen Sie für Ihr S5-Programm mit Hilfe der S5-Software eine aktuelle Querverweisliste an.
2. Kopieren Sie Ihre STEP 5-Programmdatei, die zugehörige Querverweisliste und bei Bedarf die Zuordnungsliste in ein DOS-Verzeichnis.

5.2 Operanden prüfen

Funktionsumfang der CPU

Sie müssen eventuell das zu konvertierende Programm an die S7-CPU, die Sie einsetzen wollen, anpassen.

Um sich einen Überblick über den Funktionsumfang der S7-CPU zu verschaffen, gehen Sie folgendermaßen vor:

1. Stellen Sie fest, welche S7-CPU Sie verwenden wollen.
2. Suchen Sie die S7-CPU in den Tabellen zu den Leistungsmerkmalen im Abschnitt 2.2.1 und vergleichen Sie
   - Anzahl der Operanden
   - Anzahl der Bausteine

mit den verwendeten Operanden und Bausteinen.

oder

1. Öffnen Sie den SIMATIC-Manager.
2. Wählen Sie S7-CPU in der Online-Ansicht der Projektstruktur aus.
3. Mit dem Menübefehl **Zielsystem > Baugruppenzustand** öffnen Sie einen Registerdialog, der Ihnen u.a. folgende Information bietet:
   - Im Register **Allgemein** können Sie den CPU-Typ identifizieren, den Speicherausbau entnehmen und die Größe der zur Verfügung stehenden Operandenbereiche ablesen.
   - Im Register **Bausteine** sind Informationen über die zur Verfügung stehenden Bausteine enthalten. Insbesondere sind die maximale Anzahl und Länge der Bausteinarten angegeben und alle auf der CPU vorhandenen OBs, SFBs und SFCs aufgelistet.

Anpassen des zu konvertierenden Programms

Um das zu konvertierende AWL-Programm später auf der eingesetzten S7-CPU ablauffähig zu machen, überprüfen Sie dieses auf die zulässige Anzahl an Bausteinen und Operanden und ändern es gegebenenfalls ab.

5.3 S5-Programm vorbereiten

Bereits vor dem Konvertieren können Sie Ihr STEP 5-Programm an den zukünftigen Einsatz als STEP 7-Programm vorbereiten (müssen es aber nicht; alle Korrekturen können Sie auch nach dem Konvertieren in der STEP 7-Quelldatei durchführen). Mit dieser Anpassung reduzieren Sie die Anzahl der Fehlermeldungen und Warnungen.

Beispielsweise können Sie folgende Anpassungen vor dem Konvertieren vornehmen:

- Systemeinstellungen aus Datenbausteine mit Programmeigenschaften DB 1 bzw. DX 0 auswerten und anschließendes Löschen der DB 1 bzw. DX 0.
- Entfernen aller Aufrufe von integrierten Bausteinen oder Zugriffen auf den Systemdatenbereich BS, deren Funktionalität über die Parametrierung der S7-CPU erreicht werden kann.
- Anpassen der Operandenbereiche Eingänge, Ausgänge und Peripherie an die (neuen) Baugruppenadressen mit der STEP 5-Funktion Umverdrahten (Sie sollten hierbei beachten, daß der STEP 5-Adreßbereich nicht überschritten wird, sonst wird bereits ein Fehler im ersten Konvertierlauf gemeldet; eine Konvertierung für diese Anweisungen findet dann nicht statt.).
- Sie können nichtkonvertierbare Programmteile, die wiederholt vorkommen, bis auf eine “eindeutige” STEP 5-Anweisung pro Programmteil löschen. Dieser “eindeutigen” Anweisung weisen Sie ein Makro zu (siehe Abschnitt 5.4), das den Programmteil ersetzen soll.
- Wenn Ihr Programm sehr viele (und lange) Datenbausteine enthält, die keine Datenstruktur aufweisen (z. B. als Datenpuffer verwendet werden), können Sie die Datenwörter in diesen Datenbausteinen bis auf ein Datenwort löschen. Nach dem Konvertieren (und noch vor dem Übersetzen) programmieren Sie den Inhalt dieser Datenbausteine in der Quelldatei mit einer Feld-Deklaration, z. B. Puffer: ARRAY [1..256] of WORD.

Mit dem Konverter können Sie nicht nur komplette Programme konvertieren, sondern auch einzelne Bausteine.

5.4 Makros erstellen

Nutzen

Für die Konvertierung können Sie im S5/S7 Konverter Makros definieren, für

- S5-Befehle, die nicht automatisch konvertiert werden und
- S5-Befehle, die Sie unterschiedlich zur Standardumsetzung konvertieren wollen.

Makros erweisen sich als nützlich, wenn Ihr Programm mehrere S5-Befehle enthält, auf die die oben genannten Eigenschaften zutreffen.

Makro-Funktion

Makros können ersetzen:

- S5-Befehle (Operatoren) und
- S5-Organisationsbausteine (OB).

Die Makros werden in der Datei S7S5CAPA.MAC für den Befehlssatz SIMATIC und in der Datei S7S5CAPB.MAC für den Befehlssatz International abgelegt. Arbeiten Sie mit beiden Befehlssätzen, müssen Sie die Makros in jeder Datei extra angeben. Man unterscheidet zwischen Befehlsmakros und OB-Makros. Sie können je 256 Befehlsmakros und OB-Makros erstellen.

5.4.1 Befehlsmakros

Befehlsmakros müssen folgendermaßen aufgebaut sein:

```
$MAKRO: <S5-Befehl>
S7-Befehlsfolge
$ENDMAKRO
```

Bei der Makrodefinition geben Sie für <S5-Befehl> den vollständigen Befehl (Operator und absoluter Operand) an.

Tabelle zeigt ein Makro für den Befehl E DB 0, mit dem in S5 Datenbausteine eingerichtet werden. Die Länge (in Worten) des einzurichtenden Datenbausteines steht in AKKU 1. In S7 wird die Funktion mit der Systemfunktion SFC 22 CREAT_DB realisiert. Die Länge des Datenbausteines muß in Anzahl Bytes umgerechnet werden.

Tabelle 5-1 Beispiel für ein Befehlsmakro

Makro	S5	S7
$MAKRO: E DB 0 //Ersetzt Befehl zum //Einrichten von DB	L Konstante B MW 100	L Konstante;
SLW 1 //Anzahl Worte in Anzahl //Bytes umrechnen T MW 102 CALL SFC 22(// Aufruf SFC CREAT_DB LOW_LIMIT := MW 100, UP_LIMIT := MW 100, COUNT := MW 102, RET_VAL := MW 106, DB_NUMBER := MW 104); $ENDMAKRO	E DB 0	SLW 1; T MW 102; CALL SFC 22(LOW_LIMIT := MW 100, UP_LIMIT := MW 100, COUNT := MW 102, RET_VAL := MW 106, DB_NUMBER := MW 104);

5.4.2 OB-Makros

Wegen der Unterschiede bei den Organisationsbausteinen zwischen S5 und S7 kann es für Sie empfehlenswert sein, die Umsetzung Ihrer S5-OBs selbst zu steuern. OB-Makros müssen folgendermaßen aufgebaut sein:

```
$OBCALL: <Nummer des OB>
CALL <S7-Systemfunktion>;
$ENDMAKRO
```

Wird in der S5-Quelldatei ein Befehl mit dem Operand OB x gefunden, wird dieser Befehl durch die definierten Makrobefehle ersetzt. Ausgenommen sind FB-Aufrufe, die OBs als Formalparameter verwenden.

Tabelle 5-2 Beispiel für OB-Makro

Makro	S5	S7

`$OBCALL: 31      //Ersetzt Befehle mit OB 31`
`CALL SFC 43;`
`$ENDMAKRO`
	`SPA OB 31`	`CALL SFC 43;`

Erstellungs-hinweise

Die Funktionen der Organisationsbausteine bei S5 unterscheiden sich von den Funktionen der OBs in S7. OBs, die nicht automatisch konvertiert werden, müssen Sie über die Nachbearbeitung ersetzen durch:

- OBs mit verändertem Funktionsumfang,
- neue S7-Befehle oder
- Systemeinstellungen, die Sie bei der Hardware-Parametrierung festlegen.

Ausführliche Hinweise darauf, wie S5-OBs zu ersetzen sind, finden Sie im Abschnitt 3.7.5.

Hinweis

Es wird nicht geprüft, ob ein Makro doppelt definiert wird. Ist dies der Fall, wird der zuerst definierte Makro verwendet. Die angegebene S7-Befehlsfolge wird nicht auf Richtigkeit überprüft. Achten Sie auf die korrekte Schreibweise der Schlüsselwörter und Sonderzeichen (Doppelpunkt).

5.4.3 Editieren von Makros

Makros erstellen Sie auf folgende Weise:

- Starten Sie den S5/S7 Konverter über die Schaltfläche ”Start” auf der Task-Leiste in Windows 95 über ”Simatic/STEP 7/S5-Datei konvertieren”.
- Wählen Sie den Menübefehl **Bearbeiten > Ersetzungsmakro** (Es ist keine Programmdatei geöffnet!).

Ergebnis: Die Datei S7S5CAPA.MAC wird geöffnet.

- Geben Sie die Makros wie oben beschrieben ein und sichern Sie die Datei mit dem Menübefehl **Datei > Speichern**.
- Schließen Sie die Datei mit dem Menübefehl **Datei > Schließen**.

Ergebnis: Die Datei S7S5CAPA.MAC wird geschlossen. Ab dem nächsten Konvertierlauf werden die definierten Makros gültig.

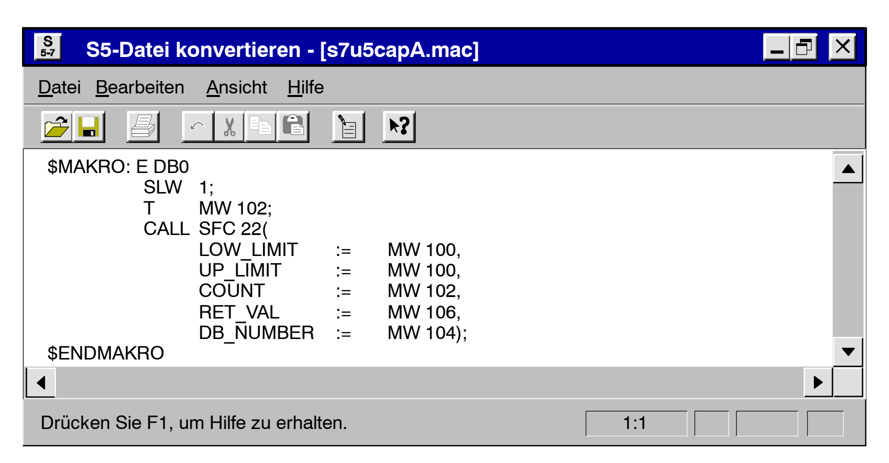


Bild 5-1 Makro im Fenster von ”S5-Datei konvertieren”

Konvertierung

6

6.1 Starten der Konvertierung

Voraussetzung

Vergewissern Sie sich vor dem Start der Konvertierung, daß sich die zu konvertierende S5-Datei, die Querverweisliste und evtl. die Zuordnungsliste in demselben Verzeichnis befinden (siehe Abschnitt 5.1, Bereitstellen der benötigten Dateien).

Starten des S5/S7 Konverter

Nachdem Sie die STEP 7-Software auf Ihrem PG installiert haben, starten Sie den S5/S7 Konverter über die Schaltfläche ”Start” auf der Task-Leiste in Windows 95

- Der Eintrag lautet ”Simatic/STEP 7/S5-Datei konvertieren”.

Der S5/S7 Konverter meldet sich nach dem Start mit folgender Einstiegsmaske:



Bild 6-1 Einstiegsmaske des S5/S7 Konverter

Auswählen einer Programmdatei

Um eine Programmdatei auszuwählen, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl Datei>Öffnen.
2. Wählen Sie das Laufwerk und das Verzeichnis aus, unter dem die zu konvertierenden Dateien abgelegt sind.
3. Markieren Sie die zu konvertierende Datei und klicken Sie auf ”OK”, um Ihre Auswahl zu bestätigen.

Ergebnis: Der S5/S7 Konverter zeigt Quell- und Zieldateien und eine Zuordnung von alten und neuen Bausteinnummern.

Das Bild zeigt das Dialogfeld ”S5-Datei konvertieren [<Name>ST.S5D]”.

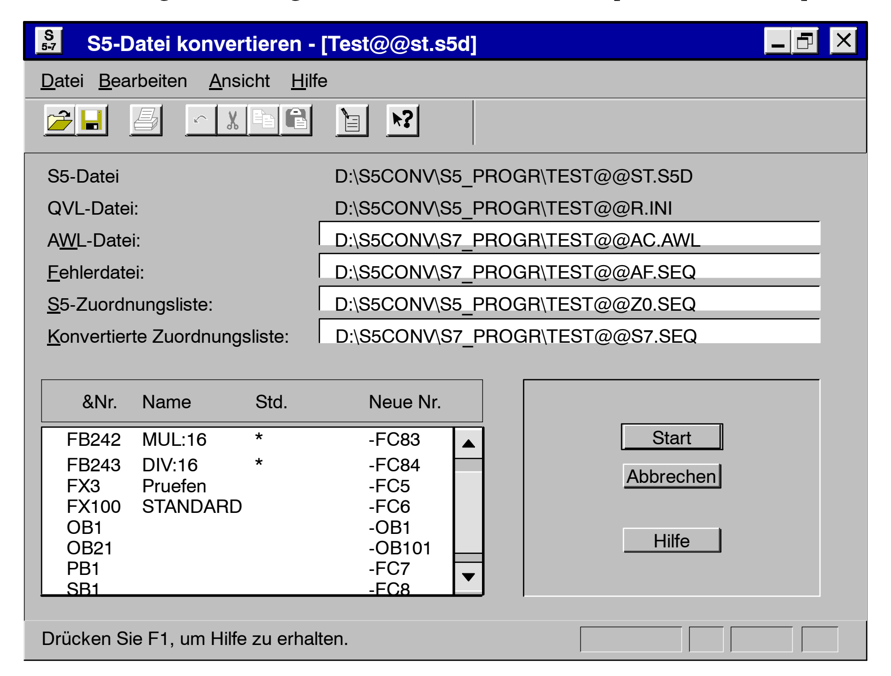


Bild 6-2 Dialogfeld ”S5-Datei konvertieren-[<Name>ST.S5D]”

Ändern der Zieldateinamen

Sie können bei Bedarf die von der Software vorgeschlagenen Namen der Zieldateien ”AWL-Datei”, ”Fehlerdatei” und ”Konvertierte Zuordnungsliste” ändern. Dies kann etwa erforderlich sein, wenn der Editor, mit dem Sie die konvertierte Datei weiterbearbeiten wollen, bestimmte Namenskonventionen verlangt (z.B. NAME.TXT).

Gehen Sie folgendermaßen vor:

- Klicken Sie in das Textfeld mit dem Pfadnamen der Zieldatei, die Sie ändern wollen.
- Ändern Sie den Text wie gewünscht.

Zuordnung Nr. -> neue Nr.

Die Software schlägt neue Nummern für die zu konvertierenden Bausteine vor und gibt sie im Dialogfeld ”S5-Datei konvertieren [<Name>ST.S5D]” an. Wenn Sie andere Nummern vergeben wollen, gehen Sie folgendermaßen vor:

1. Klicken Sie zweimal auf die Bausteinnummer, die Sie ändern wollen.
2. Geben Sie die neue Nummer in das Dialogfeld ”Neue Bausteinnummer” ein und klicken Sie auf die Schaltfläche ”OK”, um Ihre Eingabe zu bestätigen.

S5-Standardfunktionsbausteine

Enthält Ihr S5-Programm Standardfunktionsbausteine, werden diese durch ein Sternchen in der Spalte ”Std.” gekennzeichnet.

Durchführen der Konvertierung

Durch Klicken auf die Schaltfläche ”Start” stoßen Sie den Konvertiervorgang an. Der Konvertiervorgang besteht aus zwei Konvertierläufen und dem Umsetzen der Zuordnungsliste.

Im 1. Konvertierlauf wird das S5-Programm in eine S5-Quelle mit allen Bausteinen und Kommentaren umgesetzt.

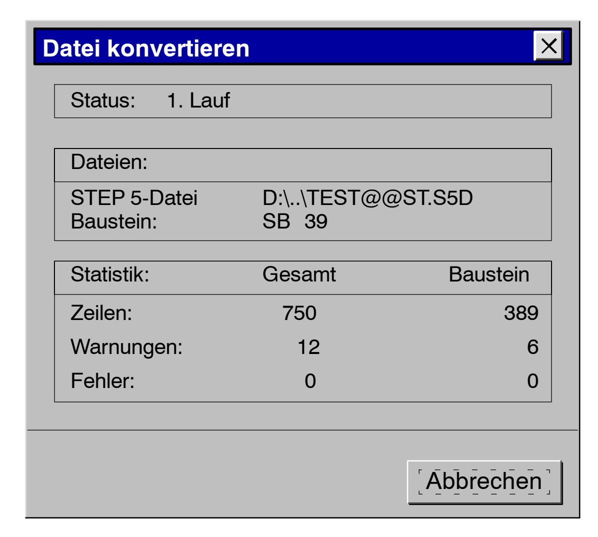


Bild 6-3 Erster Konvertierlauf

Im 2. Lauf wird die S5 Quelle in die AWL-Quell-Datei mit den neuen Bausteintypen, Bausteinnummern und S7-Syntax umgesetzt.

Zuordnungsliste umsetzen

Die Symbole der S5-Zuordnungsliste werden beim Umsetzen der Zuordnungsliste in eine vom Symbol-Editor importierbare Form konvertiert.

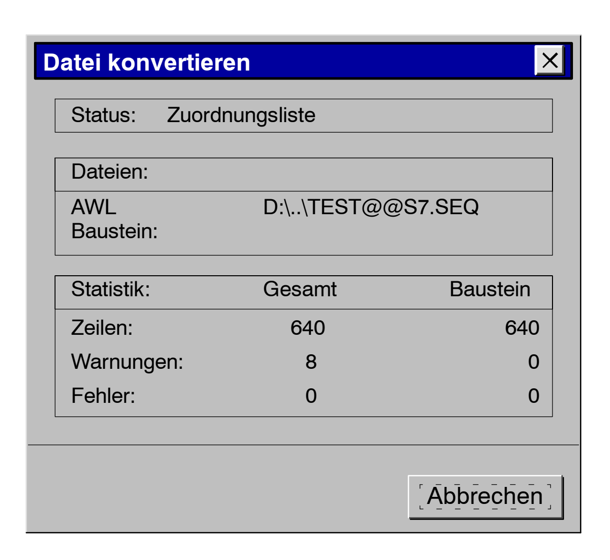


Bild 6-4 Umsetzen der Zuordnungsliste

6.2 Erzeugte Dateien

Der S5/S7 Konverter erzeugt bei der Konvertierung folgende Dateien:

- Die Datei <Name>A0.SEQ:

  Die Datei wird aus dem ersten Konvertierlauf erzeugt. Sie enthält die Datei <Name>ST.S5D in ASCII-Form.

- Die Datei <Name>AC.AWL:

  Die Datei wird aus dem zweiten Konvertierlauf erzeugt. Sie enthält das AWL-Programm. Aus diesem Lauf stammen auch eventuelle Meldungen aufgrund fehlerhafter Makrodefinitionen.

- Die Datei <Name>S7.SEQ:

  Die Datei wird aus der Umsetzung der Zuordnungsliste erzeugt. Sie enthält die konvertierte Zuordnungsliste in einer für den Symbol Editor importierbaren Form.

- Die Fehlerdatei ”<Name>AF.SEQ”:

  Sie wird im oberen Listenfeld des Fensters ”S5-Datei konvertieren” angezeigt und enthält Fehler und Warnungen, die das konvertierte Programm enthält. Diese Meldungen werden beim ersten und zweiten Konvertierlauf und bei der Umsetzung der Zuordnungsliste erzeugt.

In einem Dialogfeld wird nach Beendigung des Konvertierlaufes die Anzahl der Fehler und Warnungen ausgegeben.

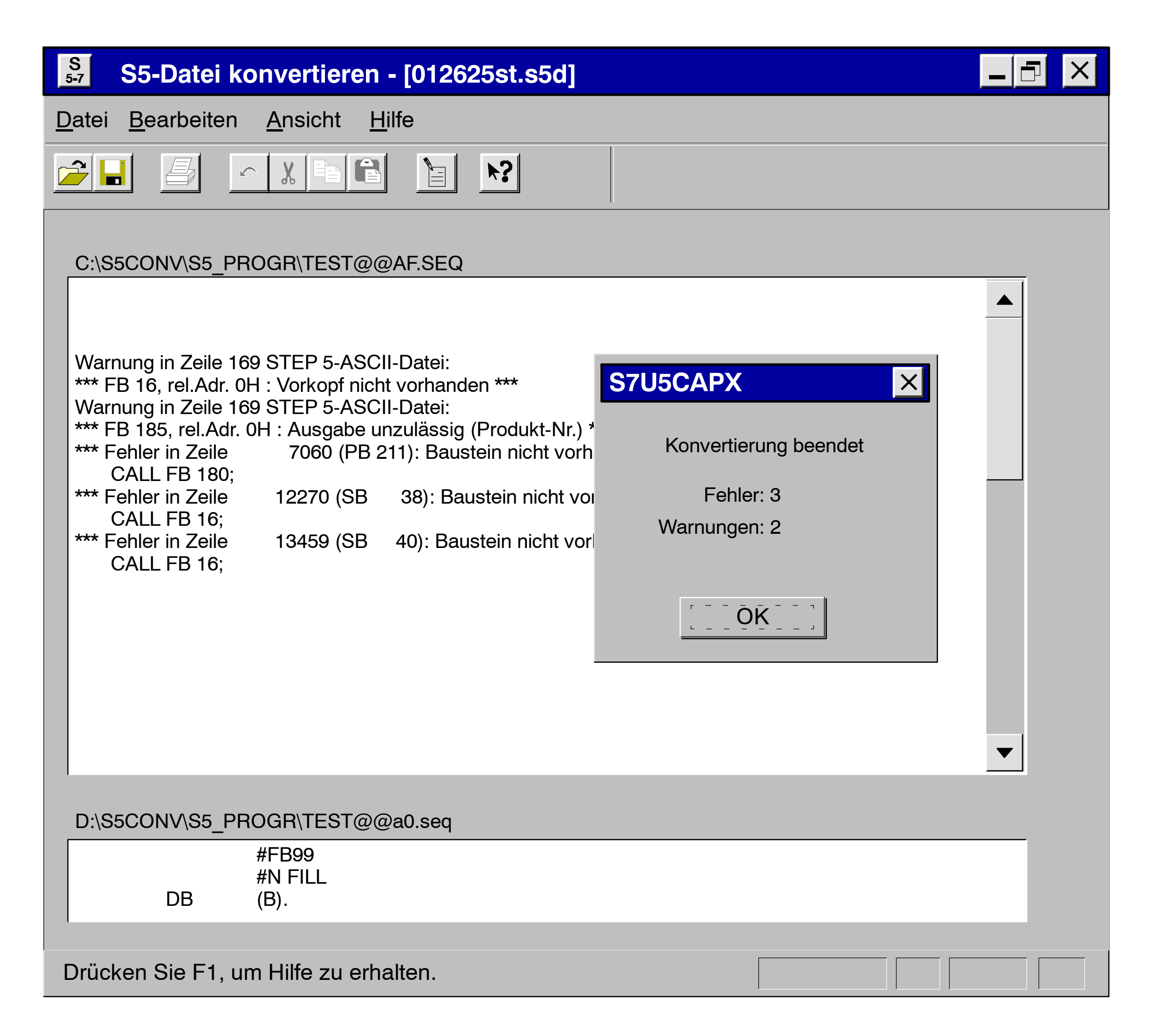


Bild 6-5 Meldungen beim Konvertieren

Lokalisieren von Fehlern

Im unteren Listenfeld des Fensters können Sie sich die Stelle in der jeweiligen Datei anzeigen lassen, an der der Fehler aufgetreten ist.

In der AWL-Quelldatei werden an den entsprechenden Stellen im Programm, an denen Fehler festgestellt wurden, Meldungen ausgegeben. Außerdem enthält die Datei Warnungen bzw. Hinweise auf eventuell entstehende Probleme (z.B. durch Änderungen in der Befehlssemantik).

Drucken von Meldungen

Mit dem Menübefehl **Datei > Drucken** können Sie die angelegten Dateien Ihrer Wahl ausdrucken.



Bild 6-6 Dialogfeld ”Drucken”

6.3 Auswerten von Fehlermeldungen

Analysieren der Meldungen

Die Meldungen beim Konvertieren umfassen Fehlermeldungen und Warnungen. Gehen Sie bei der Analyse der Meldungen folgendermaßen vor:

- Lassen Sie sich im unteren Listenfeld des Fensters ”Meldungen” die Datei anzeigen, in der der Fehler aufgetreten ist.
- Entnehmen Sie die Bedeutung der Meldung der Online-Hilfe.
- Korrigieren Sie den Fehler nach den vorgeschlagenen Abhilfemaßnahmen.

Fehlermeldungen

Fehlermeldungen werden ausgegeben, wenn Teile des S5-Programms nicht konvertierbar sind und nur als Kommentar in das S7-Programm aufgenommen werden. In der folgenden Tabelle sind alle Fehlermeldungen, deren Bedeutung und mögliche Abhilfemaßnahmen aufgelistet.

Hinweise auf Regeln

Kapitel 3 (Software) enthält die Regeln für die Konvertierung von S5-Programmen nach S7. Hier finden Sie weitere Hinweise auf mögliche Fehlerquellen und Hilfen für die Nachbearbeitung des AWL-Programms.

Tabelle 6-1 Fehlermeldungen, Bedeutung und Abhilfe

Fehlermeldung	Ursprung	Bedeutung	Abhilfe
Absoluter Parameter paßt nicht zu OPKZ	1. Lauf	Falsches Operandenkennzeichen	Überprüfen Sie den Befehl.
Baustein nicht vorhanden	1. Lauf	Aufgerufener Baustein (FB, FX) fehlt oder Baustein ist in der Bausteinliste aufgeführt, aber in der Programmdatei nicht vorhanden.	Überprüfen Sie die Programmstruktur.
	2. Lauf	Baustein wird aufgerufen, der in der Programmdatei nicht vorhanden ist.	Überprüfen Sie, ob bei der Konvertierung die Querverweisliste angegeben wurde oder überprüfen Sie die Programmstruktur.
Befehl im Baustein nicht erlaubt	1. Lauf	z.B. Sprung innerhalb eines Programmbausteins	Überprüfen Sie den Befehl.
Befehl nicht definiert	1. Lauf	Ungültiger MC5/AWL-Befehl	Korrigieren Sie die S5-Programmdatei.
	2. Lauf	Befehl in S7 nicht vorhanden	Editieren Sie einen Makro oder ersetzen Sie den Befehl durch die entsprechende S7-Befehlsfolge.
Bitzugriff auf T/Z nicht mehr möglich (Bitte überprüfen)	2. Lauf	S5-Programm enthält Bitzugriffe auf Timer und Zähler.	Überprüfen Sie das AWL-Programm.
CALL OB ist nicht erlaubt	2. Lauf	Der Aufruf von OBs ist in S7 nicht erlaubt.	Setzen Sie gegebenenfalls den Befehl CALL SFC ein.
CALL SFC xy generiert, bitte Parameterliste ergänzen	2. Lauf	Parameter für SFC fehlen.	Vervollständigen Sie die SFC-Parameterliste.

Tabelle 6-1 Fehlermeldungen, Bedeutung und Abhilfe, Fortsetzung

Fehlermeldung	Ur-sprung	Bedeutung	Abhilfe
Datei nicht vorhanden *(Fortsetzung nächste Seite)*	allge-mein	Angewählte Datei ist nicht vorhanden.	Überprüfen Sie die Programmdatei.
Falsche Klammerungstiefe	1. Lauf	Klammerungsabschluß nicht ausgeglichen.	Beachten Sie die Klammerebenen, beseitigen Sie den Programmier-fehler.
Falscher Operand	1. Lauf	Operand paßt nicht zum Befehl.	Überprüfen Sie die S5-Quelle.
	2. Lauf	Operand paßt nicht zum Befehl.	Ändern Sie die AWL-Datei.
Fehler beim Umwandeln	2. Lauf	BI ohne Konstante	Ergänzen Sie den Ladebefehl mit Konstante.
Fehler in Makrodatei, Makro xy ignoriert	2. Lauf	Makrofehler	Überprüfen Sie die Makro-anweisung
Formalparameter nicht defi-niert	1. Lauf	Mehr Parameter als im aufrufenden Baustein	Überprüfen Sie die S5-Programm-datei.
Inhaltsverzeichnis nicht vor-handen	1. Lauf	Programmdatei enthält keine Bausteine.	Überprüfen Sie die Programmdatei.
Kommentarlänge falsch	1. Lauf	Fehler in S5-Datei	Überprüfen Sie die Programmdatei.
Kommentar zu lang	1. Lauf	Fehler in S5-Datei	Überprüfen Sie die Programmdatei.
Kein Bausteinname angegeben	1. Lauf	Bausteinname besteht nur aus Blanks.	Geben Sie einen Bausteinnamen ein.
Keine Zugriffsrechte	allge-mein	Datei ist schreibgeschützt.	Heben Sie den Schreibschutz auf.
Marke nicht definiert	1. Lauf	Sprungmarke ist im Vorkopf nicht definiert.	Überprüfen Sie die S5-Datei.
Marke ungültig	1. Lauf	Sprungmarke enthält ungültige Zeichen.	Überprüfen Sie die S5-Datei.
Operator ungültig	1. Lauf	Operator in S5-Datei nicht bekannt oder nicht konvertierbar	Ersetzen Sie den Operator durch den entsprechenden S7-Befehl.
Operator ungültig, kann eventuell durch die Anweisung \"L P# Formalparameter\" ersetzt werden	2. Lauf	Der Operator kann in dieser Form nicht in S7 geladen werden.	Beutzen Sie eventuell die angegebene Anweisung.
Parameteranzahl falsch	1. Lauf	Fehler im S5-Programm	Überprüfen Sie die Programmdatei.
Parameter fehlerhaft	1. Lauf	Fehler im S5-Programm	Überprüfen Sie die Programmdatei.
Parametertyp falsch	1. Lauf	Fehler im S5-Programm	Überprüfen Sie die Programmdatei.
Schreibfehler Diskette	allge-mein	Datei ist schreibgeschützt oder es ist kein Platz auf der Diskette.	Heben Sie den Schreibschutz auf, oder löschen Sie nicht benötigte Daten.
Speicherüberlauf im PG (Platzproblem)	1. Lauf	Zuwenig Hauptspeicherplatz	Löschen Sie nicht mehr benötige Dateien aus dem Hauptspeicher.
Sprungmarke konnte nicht generiert werden	2. Lauf	SPR-Befehl über Bausteingrenze	Beseitigen Sie den Fehler im S5-Programm.
Ungültiger MC5-Code wurde konvertiert	1. Lauf	Konvertierung eines älteren S5-Befehls.	keine

Warnungen

Warnungen werden ausgegeben, wenn Teile des S5-Programms konvertiert werden, aber nochmals auf Richtigkeit überprüft werden sollten.

Tabelle 6-2 Warnungen, Bedeutung und Abhilfe

Warnung	Ursprung	Bedeutung	Abhilfe
Ausgabe unzulässig (Produkt-Nr.)	1. Lauf	S5-Standardfunktionsbaustein muß durch S7-FC ersetzt werden.	keine
Ausgabe unzulässig (GRAPH5-Baustein)	1. Lauf	GRAPH5-Bausteine sind nicht konvertierbar.	Setzen Sie eventuell einen mit GRAPH für S7 erstellten Baustein ein.
Bitte Einstellungen des Zeitrasters überprüfen	2. Lauf	Das Zeitraster kann bei S7 feiner eingestellt werden als bei S5.	Stellen Sie das Zeitraster mit der Funktion "Hardware konfigurieren" ein.
I/D beeinflußt nur Akku1-L, jetzt gesamter Akku1	2. Lauf	S7-Akkus sind auf 32 Bit erweitert.	Überprüfen Sie die Konsequenzen eines indirekten INKREMENT/ DEKREMENT-Befehls im AWL-Programm.
Neue Numerierung der Bausteine beachten	2. Lauf	Indirekter Bausteinaufruf berücksichtigt neue Bausteinnummern nicht (Nummer wird aus entsprechendem Merker- oder Datenwort geholt)	Ändern Sie die Logik in S5 oder verwenden Sie fixe Bausteinaufrufe.
OB 23 und OB 24 werden auf OB 122 konvertiert	2. Lauf	OB 23 und OB 24 werden beide in S7 durch OB 122 ersetzt.	Fassen Sie den Inhalt der OBs 23 und 24 in einem OB 122 zusammen und löschen Sie den anderen OB 122.
OB wurde als OB 34 aus AG 115U interpretiert	2. Lauf	Abhängig von der eingesetzten CPU, kann der OB 34 unterschiedliche Bedeutung haben.	Überprüfen Sie, ob dieser OB in Ihr Programm paßt.
S5-Masken-DB wird nicht zur Parametrierung von S7 verwendet.	1. Lauf	In DW0 und DW1 steht MASK.	Parametrieren Sie das AS mit STEP 7.
Sprungbefehl nach BEARBEITE nicht übersetzbar	2. Lauf	BEARBEITE-Befehl mit SPA kann nicht automatisch konvertiert werden.	Ersetzen Sie den Befehl in der AWL-Datei mit SPL und überprüfen Sie den Sprung.
Systemeinstellungen werden nicht vom S5/S7 Konverter gesetzt.	2. Lauf	DB und DX werden zwar konvertiert, haben aber nicht die gleiche Bedeutung wie in S5.	Nehmen Sie die Systemeinstellung in der Konfigurationstabelle vor.
Unterschiedliche STOP-Befehle berücksichtigen	2. Lauf	Es wird nicht zwischen STP, STS und STW unterschieden.	Überprüfen Sie die Programmdatei.
VKE wird gesetzt	2. Lauf	Bei den S5-Befehlen SU und RU wird in S7 das VKE gesetzt.	Fügen Sie bei Bedarf den Befehl CLEAR ein.
Vorkopf nicht vorhanden	1. Lauf	Für FB und FX fehlen die Bezeichnungen der Sprungmarken, für DB und DX fehlen die Datenformate.	Überprüfen Sie, ob die Vorköpfe in einer anderen Datei vorhanden sind.
Wenn AG 115U, dann auf OB 100 ändern	2. Lauf	Der Anlauf-OB 21 von S5 wird automatisch in den OB 101 konvertiert.	Lief das S5-Programm auf einem AG 115U, müssen Sie den OB 101 in OB 100 ändern.

Nachbearbeitung des konvertierten Programms 7

Nachbearbeitung vorbereiten

Für die Nachbearbeitung der erzeugten AWL-Quelldatei sind folgende Vorbereitungsschritte notwendig:

- Erzeugen Sie einen Ausdruck der Meldungen.
- Legen Sie im SIMATIC Manager in einem Projekt ein S7-Programm an, falls Sie noch kein S7-Programm in einem Projekt eingerichtet haben.
- Importieren Sie mit dem Menübefehl **Einfügen > Externe Quelle** das erzeugte AWL-Quell-Programm in den Behälter ”Quellen” des angelegten S7-Programms.
- Öffnen Sie die konvertierte Datei.

Nachbearbeitung durchführen

Für die Nachbearbeitung der erzeugten AWL-Quelldatei empfehlen wir Ihnen folgende Vorgehensweise:

- Gehen Sie im interaktiven Mode durch das Programm und ändern / ergänzen Sie anhand der Warnungen die nichtkonvertierbaren S5-Befehle und Organisationsbausteine (siehe Teil 1).

7.1 Adressenänderungen

Im Wesentlichen sind Ein- und Ausgabebaugruppen von Adressenänderungen betroffen. Die Adressen der Baugruppen können Sie HWKonfig entnehmen.

7.1.1 Möglichkeiten der Adressenänderung

Umverdrahten bei S5

Sie können die Adressen einzelner Operanden bereits vor der Konvertierung in S5 mit der Funktion Umverdrahten an die neuen S7-Adressen anpassen.

Umverdrahten bei S7

Im SIMATIC-Manager gibt es eine Funktion zum automatischen Umverdrahten für Ihre aus der Quelldatei erzeugten Bausteine.

Vorgehensweise

1. Markieren Sie im SIMATIC-Manager die Bausteine Ihres Programms, bei denen Sie Umverdrahtungen durchführen wollen.
2. Öffnen Sie die Tabelle zum Umverdrahten mit dem Menübefehl **Extras > Umverdrahten**.
3. Tragen Sie die alten und neuen Adressen der einzelnen Operanden in die Tabelle ein und speichern sie.

Die Bausteine enthalten nun die geänderten Adressen.

Adressenänderungen in der S7-Quelldatei

Passen Sie in Ihrem Programm Zugriffe auf Ein- und Ausgänge sowie direkte Peripheriezugriffe an die neuen Baugruppenadressen in S7 an.

In der S7-Quelldatei können Sie die Änderungen der Absolutadressen einfach mit dem Menübefehl **Bearbeiten > Ersetzen** durchführen.

Achtung: Es können ungewollte Änderungen entstehen, wenn sich der alte und neue Adreßbereich überlappen!

Erzeugen einer neuen (symbolisch adressierten) S7-Quelldatei

Falls Sie symbolische Adressierung verwenden wollen, können Sie die Umverdrahtung auch über die Symboltabelle durchführen.

Voraussetzung

Sie haben bereits ein fehlerfrei übersetztes Programm und eine Symboltabelle, die alle Symbole für zu ändernde Absolutadressen enthält.

Vorgehensweise

Um nun die Adressen zu ändern, gehen Sie wie folgt vor:

- Öffnen Sie einen Baustein, in dem zu ändernde Adressen vorhanden sind, und stellen in dem Menü **Extras > Einstellungen** im Register **Editor** die Option **Symbolische Darstellung** ein.
  Dies wiederholen Sie für alle Bausteine, die Adressen enthalten, die Sie ändern wollen.
- Generieren Sie aus den Bausteinen eine Quelle mit dem Menübefehl **Datei > Quelle generieren**. Die Bausteine können Sie in einem Dialogfenster auswählen, nachdem Sie den Namen der Quelle eingegeben haben.

Beachten Sie bei der Erstellung der Bausteinreihenfolge die Aufruf-Hierarchie! Es gilt prinzipiell, daß aufgerufene Bausteine bereits existieren müssen, d. h. sie müssen in der Quelle vor die Bausteine eingefügt werden, von denen sie aufgerufen werden.

Ergebnis: In der erzeugten Quelle stehen die Anweisungen mit symbolischer Adressierung.

- Jetzt können Sie in der Symboltabelle die Umverdrahtung durchführen: Ersetzen Sie die S5-Adressen, die sich geändert haben, durch die neuen S7-Adressen.
- Nach dem Übersetzen der Quelldatei sind in den Bausteinen die neuen Adressen enthalten.

7.2 Nichtkonvertierbare Funktionen

Operanden und Operationen, die nicht konvertiert werden können, werden in das erzeugte S7-Programm nur als Kommentar aufgenommen und müssen von Ihnen überarbeitet werden.

Sie haben als Anwender zwei Möglichkeiten der Umsetzung:

- Sie definieren für diese Operanden und Operationen (soweit sie im Anwenderprogramm auftreten) eigene S7-AWL Anweisungsfolgen (Makros), die dann bei der Umsetzung benutzt werden.
- Sie editieren die entsprechenden Anweisungsfolgen im resultierenden S7-Programm.

Die Entscheidung hängt u.a. davon ab, wie oft ein solcher Befehl in Ihrem Anwenderprogramm auftritt.

In den Abschnitten 3.11 und 3.12 können Sie die nichtkonvertierbaren Operanden und Operationen nachlesen. Es sind ebenso Vorschläge enthalten, wie Sie nichtkonvertierbare Funktionen in S7 realisieren können.

7.3 Indirekte Adressierung - Konvertierung

Der S5/S7 Konverter setzt die indirekte Adressierung mit B MW und B DW mit STEP 7-Anweisungen um. Die erzeugte Anweisungsfolge ist meist sehr umfangreich, da der STEP 5-Zeiger in STEP 7-Format umgerechnet werden muß und dabei ein Zwischenspeichern der Akkumulatorinhalte und des Statusworts notwendig ist.

Falls die indirekte Adressierung sehr häufig in Ihrem Programm enthalten ist, lohnt sich eine Anpassung an die indirekte Adressierung in STEP 7. Durch geeignete Programmierung kann viel Speicherplatz gespart werden.

Die Auflistung zeigt, wie der S5/S7 Konverter die indirekte Adressierung in den verschiedenen Fällen umsetzt:

Zeiten und Zähler

Die indirekte Adressierung von Zeiten und Zählern wird in speicherindirekte Adressierung unter Verwendung eines temporären Lokaldatenworts umgesetzt.

Bausteine

Die indirekte Adressierung von Bausteinen wird in speicherindirekte Adressierung unter Verwendung eines temporären Lokaldatenworts umgesetzt.

Die neuen Bausteinnummern können bei der Konvertierung nicht berücksichtigt werden und müssen deshalb korrigiert werden.

Operanden

Die indirekte Adressierung von Operanden wird bit- und wortweise in registerindirekte Adressierung unter Verwendung des Adreßregisters AR1 und temporären Lokaldaten als Zwischenspeicher für das Statuswort STW, AKKU 1 und AKKU 2 umgesetzt.

Indirekte Adressierung über das BR-Register

Die Anweisungen werden nicht konvertiert. Die indirekte Adressierung muß in S7 neu programmiert werden.

Sonstige indirekte Adressierung

Die Anweisungen müssen in S7 neu programmiert werden.

Weitere Information zur indirekten Adressierung sind in Abschnitt 3.13.4 enthalten.

7.4 Arbeiten mit direkten Speicherzugriffen

In STEP 5 wurden für einige Funktionen Zugriffe auf absolute Speicheradressen verwendet; diese Zugriffsmöglichkeiten gibt es in STEP 7 nicht mehr.

STEP 5	STEP 7
Adressierung von Datenoperanden in "überlangen" Datenbausteinen	Adressierung von Datenoperanden mit Adresse größer 255 ist jetzt mit den normalen Anweisungen (L, T, ...) möglich.
Indirekten Adressierung mit dem BR-Register	Die indirekte Adressierung kann mit der registerindirekten Adressierung realisiert werden (siehe *Registerindirekte Adressierung* in Abschnitt 3.13.4 und *AWL-Handbuch* **/232/**).
Verwendung des Blocktransfers	Für den Blocktransfer gibt es eine Systemfunktion SFC 20 BLKMOV. Die zu kopierenden Speicherbereiche werden an den Bausteinparametern angegeben. Sollen die Speicherbereiche variabel sein, können an den Parametern ANY-Pointer angegeben werden, die im Anwenderprogramm belegt werden können.

7.5 Parameterversorgung

S5-Befehl B<Bausteinparameter>

Der Befehl B <Formalparameter der Art "B"> läuft in S5 je nach übergebenem Bausteintyp ab als

- "SPA Codebaustein" oder als
- "A DB Datenbaustein".

Wegen der fehlenden Typinformation im Formalparameter ist in diesem Fall eine automatische Konvertierung nicht möglich. Untersuchen Sie also Ihr Programm auf X-Befehle mit Parametern der Art "B" und setzen Sie Befehle manuell um.

Aktualparameter

Der S5/S7 Konverter übernimmt bei parametrierten Funktionsbausteinen die Aktualparameter an Bausteinaufrufen ohne Änderung. Wenn Sie mit einem Aktualparameter Adressen vorgegeben haben, müssen Sie diese Adreßvorgabe prüfen und gegebenfalls ändern.

Beispiele:

- Angabe einer Datenwortnummer:

  muß auf byteweise Adressierung umgerechnet werden.
- Angabe einer Peripherieadresse:

  es muß die neue Baugruppenadresse eingesetzt werden.
- Übergabe eines Bausteins:

  muß mit der neuen Bausteinnummer versehen werden.

7.6 Standardfunktionen

S5-Standardfunktionsbausteine

Sind in Ihrem S5-Programm Standardfunktionsbausteine vorhanden, werden sie

- vor der Konvertierung durch ein Sternchen in der Spalte “Std.” des Dialogfeldes “S5-Datei konvertieren-[<Name>ST.S5D]” und
- nach der Konvertierung durch Ausgabe der Meldung “Ausgabe unzulässig (Produktnr)” angezeigt.

Zum Lieferumfang der S7-Basissoftware gehören bereits konvertierte S7-Funktionen (ehemalige S5-Standardfunktionsbausteine) für Gleitpunktarithmetik, Signalfunktionen, Integrierte Funktionen, Grundfunktionen und Mathematische Funktionen mit der Bezeichnung FC 61 bis FC 125 (siehe Abschnitt 3.9).

Einfügen der FCs

Gehen Sie folgendermaßen vor, um die S7-Funktionen in Ihr S7-Programm zu integrieren:

1. Öffnen Sie das Projekt, in das Sie die Funktionen einfügen möchten.
2. Öffnen Sie die Standard-Bibliothek des SIMATIC Managers mit den konvertierten S5-Funktionen (StdLib30).
3. Kopieren Sie die benötigten S7-Funktionen aus der Standardbibliothek in das S7-Programm.

Übersetzen 8

Um das konvertierte und gegebenenfalls nachbearbeitete Programm ablauffähig zu machen, müssen Sie es mit dem AWL-Compiler übersetzen. Gehen Sie dabei genauso vor, wie beim Übersetzen einer neu erstellten Text-Datei.

Konsistenz prüfen

Mit dem Menübefehl **Datei > Konsistenz prüfen** können Sie zu jedem beliebigen Zeitpunkt die Syntax und Konsistenz der Quelldatei überprüfen, ohne dabei die Erzeugung von Bausteinen anzustoßen. Dabei wird u. a.

- die Syntax,
- die Symbolik und
- die Existenz im Programm aufgerufener Bausteine

geprüft. Sie erhalten ein Übersetzungsprotokoll, in dem der Dateiname der übersetzten Datei, die Anzahl der übersetzten Zeilen, Fehlerzahl und Warnungen angegeben werden.

Quelldatei übersetzen

Mit dem Menübefehl **Datei > Übersetzen** übersetzen Sie Ihre Quelldatei in Bausteine.

Nach der Übersetzung erscheint das Übersetzungsprotokoll. Fehler werden wie bei der Konsistenzprüfung angezeigt. Sind mehrere Bausteine in einer Quelldatei programmiert, werden nur die fehlerfreien übersetzt und gespeichert.

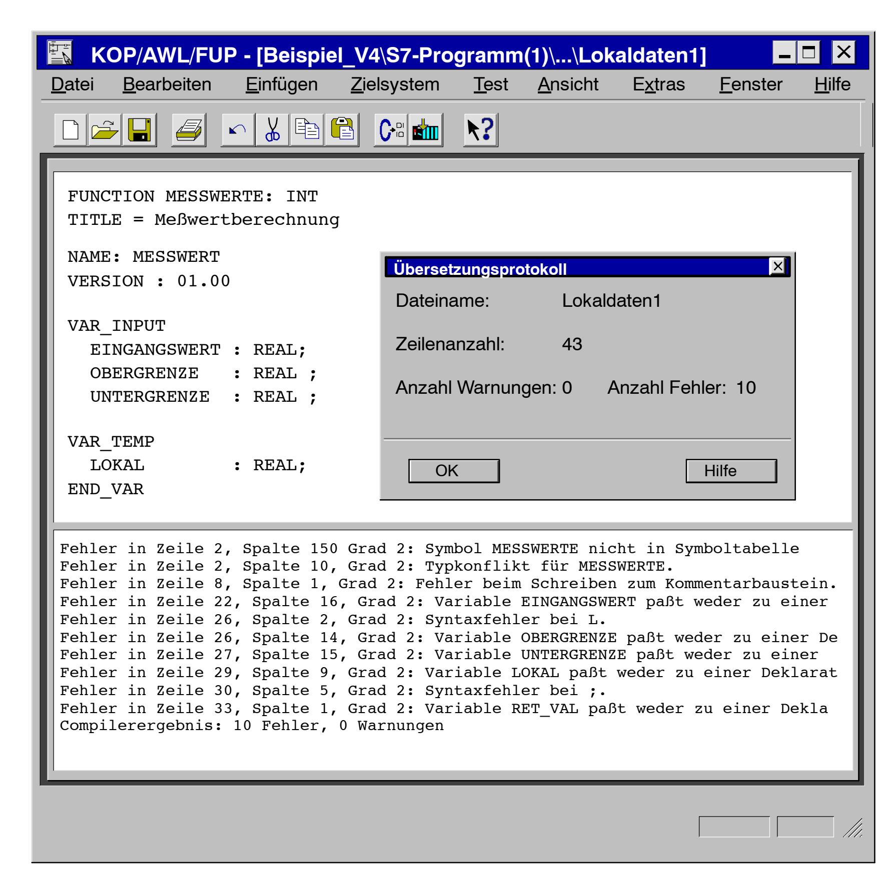


Bild 8-1 Konsistenzprüfung und Übersetzen von Quelldateien

Fehlerbehebung

Sind Fehler und / oder Warnungen in Ihrem konvertierten Programm vorhanden, werden diese nach dem Prüfen der Konsistenz oder nach dem Übersetzen unterhalb der Quelldatei in einem zweiten Teilfenster mit Angabe der Fehlerursache aufgelistet. Wenn sie eine Fehlermeldung markieren, wird Ihnen in der Quelldatei die entsprechende Fehlerstelle angezeigt. Diese Kopplung von Fehlermeldung und Fehlerstelle ermöglicht Ihnen eine schnelle Fehlerbehebung.

Fehlerkorrekturen und Änderungen können Sie im Überschreibmodus vornehmen. Zwischen Einfüge- und Überschreibmodus wechseln Sie mit der `INSERT`-Taste.

Anwendungsbeispiel 9

In diesem Kapitel werden anhand eines Beispiels vier Themenbereiche erläutert, die in S7 neu sind bzw. auf andere Art realisiert werden als in S5:

- Analogwertverarbeitung
- Lokaldaten
- Auswertung der Start-Information der Organisationsbausteine
- Blocktransfer

In dem Beispiel wird ein Motor mit Rechts- und Linkslauf über Digital-ein-/ausgabebaugruppe angesteuert. Die Drehzahl wird über eine Analogeingabebaugruppe gelesen und kann über eine Analogausgabebaugruppe ausgegeben werden. Die Digital- und Analogbaugruppen müssen für das Beispiel diagnosealarmfähig sein.

Aufbau

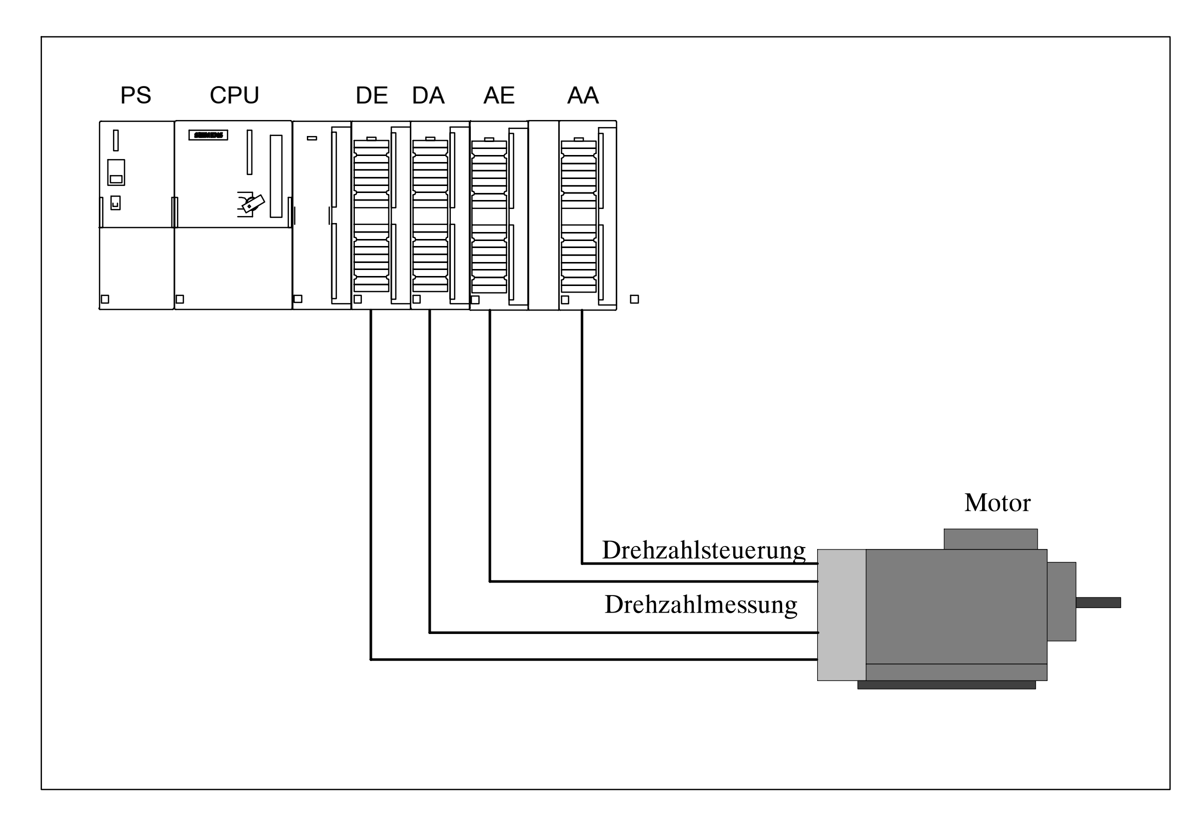


Bild 9-1 Aufbau für das Beispiel

9.1 Analogwertverarbeitung

Umwandlung von Analogwerten

Die Analogwerte werden nur in digitaler Form von der CPU verarbeitet.

Analogeingabebaugruppen wandeln das analoge Prozeßsignal in eine digitale Form um.

Analogausgabebaugruppen wandeln den digitalen Ausgabewert in ein Analogsignal um.

Analogwertdarstellung in S5

Tabelle 9-1 Beispiel für die Analogeingabebaugruppe 6ES5 460-7LA13

Auflösung	Analogwert															
Bitnummer	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wertigkeit der Bits	VZ	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	T	F	Ü

Bei Analogausgabebaugruppen werden die Werte im 12Bit-Zweierkomplement dargestellt.

Analogeingabebaugruppen können wahlweise den Wert als 12Bit-Zahl mit Vorzeichen oder 13Bit-Zweierkomplement auswerten.

Mit dem Bit “Ü” wird der Überlauf angezeigt.

Das Bit “F” ist das Fehlerbit und wird bei einem auftretenden Fehler gesetzt (z. B. Drahtbruch, wenn parametriert).

Das Bit “T” entspricht dem Tätigkeitsbit. Wenn das Bit “0” ist, ist der angezeigte Wert gültig.

Analogwertdarstellung in S7

Der digitalisierte Analogwert ist für Eingabe- und Ausgabewerte bei gleichem Nennbereich derselbe.

Die Darstellung der Analogwerte erfolgt als Zweierkomplement.

Tabelle 9-2 Beispiel für Analogeingabebaugruppen in S7

Auflösung	Analogwert															
Bitnummer	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wertigkeit der Bits	VZ	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Das Vorzeichen (VZ) des Analogwertes steht immer im Bit Nummer 15: “0” für positive und “1” für negative Werte.

Bei S7 gibt es keine Fehlerbits.

Tritt ein Fehler auf, wird der Wert W#16#7FFF ausgegeben.

Bei diagnosefähigen Baugruppen kann im Fehlerfall ein Diagnosealarm ausgelöst werden. Die Einstellung des Diagnosealarms erfolgt in der HWKonfig.

Beträgt die Auflösung einer Analogbaugruppe weniger als 15 Bit, steht der Analogwert linksbündig in den Nutzdaten. Die nicht besetzten niederwertigen Stellen haben den Signalzustand “0”.

Beispiel

Im Beispiel wird die Drehzahl des Motors von einer Analogeingabebaugruppe gelesen. Es wird eine Analogeingabebaugruppe mit einer Auflösung von 14 Bit verwendet. Der Meßwert ist ein bipolarer Wert (z. B. Meßbereich +/-10V).

Ober- und Untergrenze werden als Parameter übergeben.

Der Analogwert wird auf Ober- und Untergrenze geprüft. Falls der gelesene Wert außerhalb des zulässigen Bereichs ist, wird ein Fehler über das Binärergebnis (BIE = “0”) gemeldet und als Wert 0 ausgegeben. Ist der Wert in Ordnung, wird der gelesene Wert ausgegeben.

Der Analogwert wird über den Rückgabewert RET_VAL der Funktion ausgegeben. Der RET_VAL stellt einen Funktionswert dar. Diese Funktionalität ist neu gegenüber S5.

```
FUNCTION FC1: REAL
TITLE = Analogwertverarbeitung
NAME:               ANALOG
VERSION:   01.00

VAR_INPUT
     EINGANGSWERT   : INT;        // Eingangswert
     OBERGRENZE     : REAL;       // Obergrenze für den Analogwert
     UNTERGRENZE    : REAL;       // Untergrenze für den Analogwert
END_VAR

BEGIN
NETWORK
TITLE = Überprüfung auf Ober- und Untergrenze

     O(;
     L      EINGANGSWERT; // Eingangswert > Obergrenze
     L      +27648;
     >I;
     );
     O(;                  // oder
     L      EINGANGSWERT; // Eingangswert < Untergrenze
     L      -27648;
     <I;
     );
     NOT;
     L      0;
     SPBNB  ENDE;         // bei Über-/Unterschreitung keine weitere
                          // Bearbeitung, Rückgabewert = 0 und BIE = ”0”
                          // keine Über-/Unterschreitung => BIE = ”1”
NETWORK
TITLE = Digitalwert in Drehzahl umrechnen

     L      OBERGRENZE;   // Formel für Umrechnung EINGANGSWERT in Drehzahl:
     L      UNTERGRENZE;  // Analogwert = (OBERGRENZE - UNTERGRENZE)
                          //              * EINGANGSWERT
     -R;                  //              / (55296 (Anzahl Einheiten))
     L      EINGANGSWERT;
     ITD;                 // Wert in Gleitpunktzahl umwandeln
     DTR;
     *R;
     L      55296.0;
     /R;
ENDE:       T       RET_VAL;
     BE;

END_FUNCTION
```

Bild 9-2 Analogwertverarbeitung

9.2 Temporäre Lokaldaten

Temporäre Lokaldaten dienen als Zwischenspeicher und sind damit ein Ersatz für die S5-Schmiermerker. Temporäre Lokaldaten können Sie in allen Codebausteinen verwenden. Nach der Bearbeitung des Codebausteins gehen diese Daten verloren. Die Daten stehen im Lokaldaten-Stack (L-Stack).

Beispiel 1

Das Beispiel 1 verwendet die temporären Lokaldaten als Zwischenspeicher, die **symbolisch** adressiert werden. Es wird eine vorgegebene Drehzahl in den digitalisierten Meßwert für die Analogausgabebaugruppe umgerechnet. Es wird eine Analogausgabebaugruppe mit einer Auflösung von 14 Bit verwendet. Der Meßwert ist ein bipolarer Wert (z. B. Meßbereich +/-10V).

Ober- und Untergrenze werden als Parameter übergeben.

Der Meßwert wird über den Rückgabewert der Funktion (RET_VAL) ausgegeben. Jede Funktion kann optional einen Rückgabewert liefern. Der Datentyp des Rückgabewertes wird bei der Bezeichnung der Funktion angegeben. Soll kein Rückgabewert geliefert werden, wird an Stelle des Datentyps VOID geschrieben.

```
FUNCTION FC2: INT
TITLE = Meßwertberechnung
NAME:               MESSWERT
VERSION:   01.00

VAR_INPUT
    EINGANGSWERT    : REAL;       // Eingangswert (Stromwert)
    OBERGRENZE      : REAL;       // Obergrenze
    UNTERGRENZE     : REAL;       // Untergrenze
END_VAR

VAR_TEMP
    LOKAL           : REAL;       // Lokaldaten als Zwischenergebnis
END_VAR

BEGIN
NETWORK
TITLE = Meßwertberechnung

    L      EINGANGSWERT;          // Formel zur Berechnung der Einheiten:
    L      55296.0;               // Meßwert = EINGANGSWERT
    *R;                           //            * 55296 (Anzahl Einheiten)
                                  //            / (OBERGRENZE - UNTERGRENZE)
    T      LOKAL;                 // Zwischenergebnis in Lokaldaten
    L      OBERGRENZE;            // zwischenspeichern
    L      UNTERGRENZE;
    -R;
    L      LOKAL;
    TAK;
    /R;
    RND;                          // Gleitpunktzahl in Ganzzahl umwandeln
    T      RET_VAL;

END_FUNCTION
```

Bild 9-3 Meßwertberechnung

Beispiel 2

Das Beispiel 2 verwendet die Lokaldaten wie S5-Schmiermerker, die **absolut** adressiert werden. Es wird eine Motoransteuerung mit Rechtslauf und Linkslauf realisiert. Im Beispiel werden das Eingangsbyte und das Ausgangsbyte in den Lokaldatenbereich kopiert. Für die Verwendung der temporären Lokaldaten muß der Anwender im L-Stack einen Bereich reservieren, da der L-Stack auch vom Programmeditor verwendet wird. Die absoluten Adressen der Lokaldaten können Sie im Baustein im Deklarationsteil nachlesen. Die Lokaldatenbits werden im Programm miteinander verknüpft. Daraus ergeben sich die Ausgangssignale, die am Ende des Bausteins von den Lokaldaten auf das Ausgangsbyte zurückgeschrieben werden. Die Adressen des Eingangs- und des Ausgangsbytes sind parametrierbar.

Hinweis

Fügen Sie vor den bereits bestehenden Lokaldaten neue Variable ein, verschieben sich die Adressen der nachfolgenden Lokaldaten!

Tabelle 9-3 Belegung der Ein- und Ausgänge / Lokaldaten

Adresse	Lokaldaten	Bezeichnung	Beschreibung
E n.0	L 0.0	EIN	Ein-Schalter
E n.1	L 0.1	STOP	Motor anhalten
E n.2	L 0.2	NOT_AUS	Not-Aus-Schalter
E n.3	L 0.3	MOTOR_RECHTS	Motor Rechtslauf einschalten
E n.4	L 0.4	MOTOR_LINKS	Motor Linkslauf einschalten
E n.5	L 0.5	ENDSCHALTER_RECHTS	Endschalter rechts
E n.6	L 0.6	ENDSCHALTER_LINKS	Endschalter links
E n.7	L 0.7	-	frei
A m.0	L 1.0	BEREIT	Motor ist bereit
A m.1	L 1.1	RECHTSLAUF	Rechtslauf aktiv
A m.2	L 1.2	LINKSLAUF	Linkslauf aktiv
A m.3	L 1.3	POSITION_ERREICHT	Position erreicht

Funktionsweise

Die Spannung wird mit dem Ein-Schalter zugeschaltet. Der Motor ist jetzt bereit, dies wird mit dem Ausgang BEREIT signalisiert. Mit den Tastern MOTOR_RECHTS und MOTOR_LINKS kann der Motor in die gewünschte Richtung gefahren werden. Der Motor kann jeweils nur in eine Richtung angesteuert werden. Bei einem Richtungswechsel muß der Motor zuerst mit STOP angehalten werden. Wird ein Endschalter betätigt, wird der Motor angehalten. Bei NOT_AUS wird der Motor ebenfalls gestoppt und kann erst wieder angesteuert werden, wenn der NOT_AUS-Schalter wieder zurückgesetzt ist.

```
FUNCTION FC3: VOID
TITLE = Motorsteuerung
NAME:      MOTOR
VERSION:   01.00

VAR_INPUT
    EINGANGSBYTE         : BYTE;  // Eingangsbyte
END_VAR

VAR_IN_OUT
    AUSGANGSBYTE         : BYTE;  // Ausgangsbyte
END_VAR

VAR_TEMP
    ABBILD_EINGANGSBYTE  : BYTE;  // Abbild des Eingangsbytes
    ABBILD_AUSGANGSBYTE  : BYTE;  // Abbild des Ausgangsbytes
END_VAR

BEGIN
NETWORK
TITLE =  Motorsteuerung

    L      EINGANGSBYTE; // Eingangsbyte in Lokaldatenbereich kopieren
    T      ABBILD_EINGANGSBYTE;
    L      AUSGANGSBYTE; // Ausgangsbyte in Lokaldatenbereich kopieren
    T      ABBILD_AUSGANGSBYTE;

    ON     L0.0;         // Motor nicht eingeschaltet (keine Spannung)
    ON     L0.2;         // oder NOT_AUS-Schalter betätigt
    R      L1.0;         // => Motor ist bereit rücksetzen
    R      L1.1;         // => Ansteuerung des Motor rücksetzen
    R      L1.2;
    R      L1.3;         // => Position erreicht rücksetzen
    SPB    ENDE;         // => keine weitere Signalauswertung

    U      L0.0;         // Motor eingeschaltet
    S      L1.0;         // => Motor ist bereit setzen

    U      L0.3;         // Ansteuerung des Motors nach Rechts
    UN     L0.4;         // Verriegelung: Keine Ansteuerung nach Links
    UN     L1.2;         // und Linkslauf nicht aktiv
    FP     M0.0;         // positive Flanke bilden
    S      L1.1;         // Dann: Rechtslauf einschalten
    R      L1.3;         // Position erreicht rücksetzen

    U      L0.4;         // Ansteuerung des Motors nach Links
    UN     L0.3;         // Verriegelung: Keine Ansteuerung nach Rechts
    UN     L1.1;         // und Rechtslauf nicht aktiv
    FP     M0.1;         // positive Flanke bilden
    S      L1.2;         // Dann: Linkslauf einschalten
    R      L1.3;         // Position erreicht rücksetzen
```

Fortsetzung auf nächster Seite

Bild 9-4 Funktion zur Motorsteuerung

```
     O(;
     U      L0.5;         // Endschalter Rechts erreicht und
     U      L1.1;         // Rechtslauf aktiv
     );
     O(;                  // oder
     U      L0.6;         // Endschalter Links erreicht und
     U      L1.2;         // Linkslauf aktiv
     );
     S      L1.3;         // => Position erreicht setzen
     O      L0.1;         // Motor anhalten betätigt oder
     O      L1.3;         // Position erreicht
     R      L1.1;         // => Ansteuerung des Motors rücksetzen
     R      L1.2;

ENDE:       L       ABBILD_AUSGANGSBYTE;      // Lokaldaten nach Ausgangsbyte
                                              // kopieren
     T      AUSGANGSBYTE;

END_FUNCTION
```

Bild 9-5   Funktion zur Motorsteuerung, Fortsetzung

9.3 Auswertung der Start-Information des Diagnosealarm-OB (OB 82)

Startinformation

Beim Aufruf der Organisationsbausteine durch das Betriebssystem wird dem Anwender eine systemeinheitliche Start-Information im Lokaldaten-Stack zur Verfügung gestellt. Die Start-Information hat eine Länge von 20 Byte und steht nach dem Start der OB-Bearbeitung zur Verfügung.

Startinformation des OB 82

Die Start-Information des Diagnosealarm-OBs enthält die logische Basisadresse sowie eine vier Byte lange Diagnoseinformation. Das Referenzhandbuch **/235/** beschreibt den genauen Aufbau der Startinformation. Vorlagen für die entsprechende Variablendeklarationstabelle stehen in der Standard-Bibliothek “StdLib30” unter “StdOBs”.

Die Digitalbaugruppen stellen eine Diagnosealarmanforderung an die CPU (sowohl bei kommendem als auch bei gehendem Ereignis), sofern Sie den Diagnosealarm beim Konfigurieren der Hardware freigegeben haben. Daraufhin ruft das Betriebssystem den OB 82 auf.

Sie können den Aufruf des Diagnosealarm-OBs mit Hilfe der SFCs 39 bis 42 sperren bzw. verzögern und wieder freigeben. Nähere Informationen hierzu entnehmen Sie dem Referenzhandbuch **/235/**.

Beispiel

Im Beispielprogramm wird die externe Hilfsspannung ausgewertet. Wird die externe Hilfsspannung unterbrochen, wird im Datenbaustein DB 82 “DB_DIAG” das Bit EXT_SPANNUNG_FEHLT gesetzt. Zusätzlich werden die Baugruppenadresse und der Zeitpunkt gespeichert. Die Information kann im weiteren Programm verarbeitet werden.

Vor dem Übersetzen der AWL-Quelle muß das Symbol für den Datenbaustein DB 82 “DB_DIAG” in die Symboltabelle eingetragen werden.

```
DATA_BLOCK DB_DIAG
TITLE = Diagnose-Daten
NAME:      DB_DIAG
VERSION:   01.00

STRUCT
    MDL_ADDR            : INT;        // Baugruppenadresse
    EXT_SPANNUNG_FEHLT  : BOOL;       // Fehlerbit Ext. Hilfsspannung fehlt
    DATE_TIME      : DATE_AND_TIME;   // Datum und Uhrzeit, zu denen der
                                      // Diagnosealarm ausgelöst wurde
    SFC_RET_VAL         : INT;        // Returncode des SFC BLKMOV
END_STRUCT;

BEGIN
END_DATA_BLOCK

ORGANIZATION_BLOCK OB82
TITLE = Diagnosealarm
NAME:      Diagnose
VERSION:   01.00

VAR_TEMP
    OB82_EV_CLASS       : BYTE;  // Ereignisklasse und Kennungen:
                                 // B#16#38: gehendes Ereignis
                                 // B#16#39: kommendes Ereignis
    OB82_FLT_ID         : BYTE;  // Fehlercode (B#16#42)
    OB82_PRIORITY       : BYTE;  // Prioritätsklasse 26 oder 28
    OB82_OB_NUMBR       : BYTE;  // OB-Nummer
    OB82_RESERVED_1     : BYTE;  // Reserviert
    OB82_IO_FLAG        : BYTE;  // Eingabebaugruppe: B#16#54
                                 // Ausgabebaugruppe: B#16#55
    OB82_MDL_ADDR       : INT;   // Logische Basisadresse der Baugruppe,
                                 // in der der Fehler aufgetreten ist
    OB82_MDL_DEFECT     : BOOL;  // Baugruppenstörung
    OB82_INT_FAULT      : BOOL;  // Interner Fehler
    OB82_EXT_FAULT      : BOOL;  // Externer Fehler
    OB82_PNT_INFO       : BOOL;  // Kanalfehler vorhanden
    OB82_EXT_VOLTAGE    : BOOL;  // Externe Hilfsspannung fehlt
    OB82_FLD_CONNCTR    : BOOL;  // Frontstecker fehlt
    OB82_NO_CONFIG      : BOOL;  // Baugruppe nicht parametriert
    OB82_CONFIG_ERR     : BOOL;  // Falsche Parameter in Baugruppe
    OB82_MDL_TYPE       : BYTE;  // Bit0-3: Baugruppenklasse
                                 // Bit4: Kanalinformation vorhanden
                                 // Bit5: Anwenderinformation vorhanden
                                 // Bit6: Diagnosealarm von Stellvertreter
                                 // Bit7: Reserve
    OB82_SUB_MDL_ERR    : BOOL;  // Anwendermodul falsch / fehlt
    OB82_COMM_FAULT     : BOOL;  // Kommunikationsstörung
    OB82_MDL_STOP       : BOOL;  // Betriebszustand (0: RUN, 1: STOP)
    OB82_WTCH_DOG_FLT   : BOOL;  // Zeitüberwachung hat angesprochen
    OB82_INT_PS_FLT     : BOOL;  // Baugruppeninterne Versorgungsspannung
                                 // ausgefallen
    OB82_PRIM_BATT_FLT  : BOOL;  // Batterie leer
    OB82_BCKUP_BATT_FLT : BOOL;  // Gesamte Pufferung ausgefallen
    OB82_RESERVED_2     : BOOL;  // Reserviert
    OB82_RACK_FLT       : BOOL;  // Baugruppenträgerausfall
    OB82_PROC_FLT       : BOOL;  // Prozessorausfall
    OB82_EPROM_FLT      : BOOL;  // EPROM-Fehler
    OB82_RAM_FLT        : BOOL;  // RAM-Fehler
```

Fortsetzung auf nächster Seite

Bild 9-6   Auswertung von Diagnosedaten

```
    OB82_ADU_FLT        : BOOL;         // ADU/DAU-Fehler
    OB82_FUSE_FLT       : BOOL;         // Sicherungsausfall
    OB82_HW_INTR_FLT    : BOOL;         // Prozeßalarm verloren
    OB82_RESERVED_3     : BOOL;         // Reserviert
    OB82_DATE_TIME      : DATE_AND_TIME; // Datum und Uhrzeit, zu denen der
                                        // OB angefordert wurde
END_VAR

BEGIN
NETWORK
TITLE = Diagnosealarm

    L      OB82_MDL_ADDR;         // Baugruppenadresse speichern
    T      DB_DIAG.MDL_ADDR;

    L      OB82_EV_CLASS;         // Ereignisklasse = B#16#38:
    L      B#16#38;                     // Gehendes Ereignis
    ==I;
    SPB    GEHE;
                                        // Kommendes Ereignis:
    U      OB82_EXT_VOLTAGE;            // Überprüfen, ob externe
                                        // Hilfsspannung fehlt
    S      DB_DIAG.EXT_SPANNUNG_FEHLT;  // Bit setzen
    SPA    ZEIT;
                                        // Gehendes Ereignis:
GEHE:      U       OB82_EXT_VOLTAGE;    // Externe Hilfsspannung wieder
                                        // vorhanden
    R      DB_DIAG.EXT_SPANNUNG_FEHLT;  // Bit rücksetzen

NETWORK

TITLE = Zeit speichern
ZEIT:      CALL    SFC 20(              // SFC BLKMOV
    SRCBLK :=OB82_DATE_TIME,            // Datum und Uhrzeit, zu der der
    RET_VAL:=DB_DIAG.SFC_RET_VAL, // Diagnosealarm angefordert
    DSTBLK :=DB_DIAG.DATE_TIME);        // wurde, speichern

END_ORGANIZATION_BLOCK
```

Bild 9-7 Auswertung von Diagnosedaten, Fortsetzung

9.4 Blocktransfer

Mit der Systemfunktion SFC 20 “BLKMOV” (block move) kopieren Sie den Inhalt eines Speicherbereiches (=Quellfeld) in einen anderen Speicherbereich (=Zielfeld).

Mit der SFC 20 “BLKMOV” können Sie alle Eingänge, Ausgänge, Merker und Daten kopieren.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SRCBLK	INPUT	ANY	E, A, M, D, L	Angabe des Speicherbereichs, der kopiert werden soll (Quellfeld).
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
DSTBLK	OUTPUT	ANY	E, A, M, D, L	Angabe des Speicherbereichs, in den kopiert werden soll (Zielfeld).

Hinweis

Quell- und Zielfeld dürfen sich nicht überlappen. Ist das angegebene Zielfeld größer als das Quellfeld, dann werden auch nur so viele Daten in das Zielfeld kopiert, wie im Quellfeld stehen.

Ist das angegebene Zielfeld kleiner als das Quellfeld, dann werden nur so viele Daten kopiert, wie das Zielfeld aufnehmen kann.

Wenn Sie die Parameter für Quell- und Zielbereich der SFC 20 “BLKMOV” nicht mit konstanten Zeigern belegen wollen, sondern die Bereiche variabel vorgeben wollen, können Sie dies mit Hilfe von temporären Variablen des Datentyps ANY.

ANY-Pointer für Datentypen

Die folgenden Tabellen zeigen den Aufbau des ANY-Pointers.

Tabelle 9-4 ANY-Pointer

Byte n	Byte n+1	Byte n+2	Byte n+3	Byte n+4	Byte n+5	Byte n+6	Byte n+7	Byte n+8	Byte n+9
B#16#10	Typ *(siehe Tabelle 9-5)*	Länge		Datenbaustein-Nr bei Datenbaustein		Bereichszeiger *(siehe Bild 9-8)*			

Tabelle 9-5 Typ (Byte n+1)

Wert:	**01**	**02**	**03**	**04**	**05**	**06**	**07**
Typ:	BOOL	BYTE	CHAR	WORD	INT	DWORD	DINT
Wert:	**08**	**09**	**0A**	**0B**	**0C**	**0E**	**13**
Typ:	REAL	DATE	TOD	TIME	S5TIME	DT	String

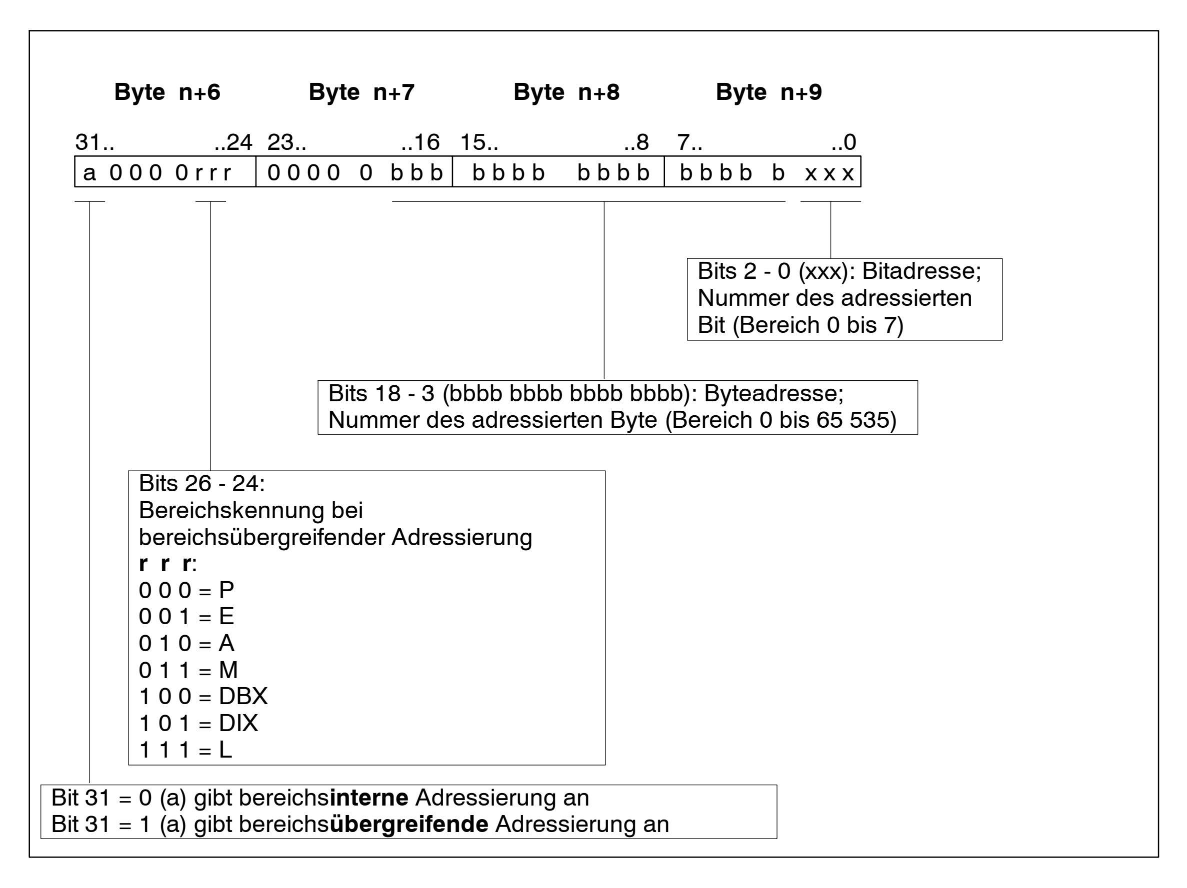


Bild 9-8 Bereichszeiger (Byte n+6 bis Byte n+9)

Beispiel

Das Beispiel enthält eine Funktion, mit der anhand der Systemfunktion SFC 20 “BLKMOV” Datenbereiche (in Datenbausteinen) kopiert werden können. Quell- und Zielbereich können variabel an den Parametern angegeben werden.

Prinzip

Die Funktion enthält zwei ANY-Pointer im Lokaldatenbereich, einen ANY-Pointer für den Quellbereich und einen ANY-Pointer für den Zielbereich. Generell gilt, daß der Datentyp ANY nur für Variable im Lokaldatenbereich zugelassen ist!

Die ANY-Pointer werden in der Funktion nach dem vorher beschriebenen Aufbau belegt und beim Aufruf des SFC 20 “BLKMOV” an den Parametern angegeben.

```
FUNCTION FC4: INT
TITLE = Kopieren von Datenbereichen
NAME:      COPY
VERSION:   01.00

VAR_INPUT
     QUELLE_DBNR     : INT;        // DB-Nr. des Quellbereichs
     QUELLE_ANFANG   : INT;        // Datenwort-Nr. des Anfangs des Quellbereichs
     QUELLE_LAENGE   : INT;        // Länge des Quellbereichs in Byte
     ZIEL_DBNR       : INT;        // DB-Nr. des Zielbereichs
     ZIEL_ANFANG     : INT;        // Datenwort-Nr. des Anfangs des Zielbereichs
     ZIEL_LAENGE     : INT;        // Länge des Zielbereichs in Byte
END_VAR

VAR_TEMP
     ZEIGER_QUELLE   : ANY;        // Any-Pointer für den Quellbereich
     ZEIGER_ZIEL     : ANY;        // Any-Pointer für den Zielbereich
END_VAR

BEGIN
NETWORK
TITLE = Quellzeiger aufbereiten

     L      P##ZEIGER_QUELLE;      // Adresse des Zeigers auf Quellbereich
     LAR1;                         // in Adreßregister 1 laden
     L      W#16#1002;             // Bereichskennung für Datenbereich in
     T      LW[AR1, P#0.0];        // Any-Pointer für Quelle schreiben
     L      QUELLE_DBNR;           // DB-Nr. in Any-Pointer für Quelle schreiben
     T      LW[AR1, P#4.0];
     L      QUELLE_ANFANG;// Anfang des Datenbereich in Pointerformat
     SLD    3;                     // umwandeln,
     OD     DW#16#84000000;        // Bereichskennung hineinverknüpfen
     T      LD[AR1, P#6.0];        // und in Any-Pointer für Quelle schreiben
     L      QUELLE_LAENGE;// Länge des Datenbereichs in Any-Pointer
     T      LW[AR1, P#2.0];        // für Quelle schreiben
```

Fortsetzung auf nächster Seite

Bild 9-9 Kopieren von Datenbereichen

```
NETWORK
TITLE = Zielzeiger aufbereiten

    L      P##ZEIGER_ZIEL;        // Adresse des Zeigers auf Zielbereich
    LAR1;                         // in Adreßregister 1 laden
    L      W#16#1002;             // Bereichskennung für Datenbereich in
    T      LW[AR1, P#0.0];        // Any-Pointer für Ziel schreiben
    L      ZIEL_DBNR;             // DB-Nr. in Any-Pointer für Ziel schreiben
    T      LW[AR1, P#4.0];
    L      ZIEL_ANFANG;           // Anfang des Datenbereich in Pointerformat
    SLD    3;                     // umwandeln,
    OD     DW#16#84000000;        // Bereichskennung hineinverknüpfen
    T      LD[AR1, P#6.0];        // und in Any-Pointer für Ziel schreiben
    L      ZIEL_LAENGE;           // Länge des Datenbereichs in Any-Pointer
    T      LW[AR1, P#2.0];        // für Ziel schreiben
NETWORK
TITLE = Daten kopieren
    CALL   SFC 20(                // Daten kopieren mit SFC BLKMOV (Blocktran-
sfer)
    SRCBLK := ZEIGER_QUELLE,      // Zeiger auf Quellfeld
    RET_VAL:= RET_VAL,            // Returncode des SFC BLKMOV
    DSTBLK := ZEIGER_ZIEL);       // Zeiger auf Zielfeld
END_FUNCTION
```

Bild 9-10 Kopieren von Datenbereichen, Fortsetzung

9.5 Aufruf der Beispiele

Dieser Abschnitt beinhaltet die Symboltabelle, die Datenbausteine, die für die Belegung der Bausteinparameter benötigt werden und den Organisationsbaustein OB 1 mit den Aufrufen der vorher beschriebenen Funktionen.

Tabelle 9-6 Symboltabelle

Symbol	Adresse	Datentyp	Kommentar
DB_DIAG	DB 82	DB 82	Diagnose-Datenbaustein
DB_MESSWERTE	DB 100	DB 100	Datenbaustein für Meßwerte
DB_MOTOR_1	DB 110	DB 110	Datenbausteine für Motor 1
FEHLER	MW 100	WORD	Rückgabewert der Funktion FC 4 für Blocktransfer

```
DATA_BLOCK DB_MESSWERTE
TITLE = Meßwerte
NAME:      DB_MESS
VERSION:   01.00
STRUCT
    ANALOGWERT_1    : REAL;       // Analogwert 1 von FC 1
    ANALOGWERT_2    : REAL;       // Analogwert 2 von FC 2
    DIGITALWERT_2   : INT;        // Digitalisierter Meßwert von FC 2
END_STRUCT;
BEGIN
END_DATA_BLOCK

DATA_BLOCK DB_MOTOR_1
TITLE = Motordaten
NAME:      DB_MOT_1
VERSION:   01.00
STRUCT
    STEUERWORT      : WORD;       // Ansteuerung von Motor 1
    DREHZAHL        : REAL;       // Drehzahl von Motor 1
    TEMPERATUR      : REAL;       // Temperatur von Motor 1
    STROM           : REAL;       // Stromverbrauch von Motor 1
END_STRUCT;
BEGIN
END_DATA_BLOCK

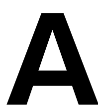
ORGANIZATION_BLOCK OB1
TITLE = Aufruf im Zyklus
NAME:              ZYKLUS
VERSION:   01.00
VAR_TEMP
  STARTINFO: ARRAY [1..20] OF BYTE;
END_VAR
BEGIN
NETWORK
TITLE = Aufruf der Funktionen
CALL FC 1(                        // Aufruf Funktion für
    EINGANGSWERT    := EW 0,      // Analogwertverarbeitung
    OBERGRENZE      := +10.0,     // Meßbereich: +/-10V
    UNTERGRENZE     := -10.0,
    RET_VAL         := DB_MESSWERTE.ANALOGWERT_1);
                                  // RET_VAL = Analogwert
                                  // Aufruf Funktion für Berechnung des
CALL FC 2(                        // digitalisierten Meßwertes
    EINGANGSWERT    := DB_MESSWERTE.ANALOGWERT_2,//
    OBERGRENZE      := +10.0,     // Meßbereich: +/-10V
    UNTERGRENZE     := -10.0,
    RET_VAL         := DB_MESSWERTE.DIGITALWERT_2);
                                  // RET_VAL = digitalisierter Meßwert
CALL FC 3(                        // Aufruf Funktion für Motorsteuerung
    EINGANGSBYTE    := EB 4,
    AUSGANGSBYTE    := AB 8);
CALL FC 4(                        // Aufruf Funktion für Blocktransfer
    QUELLE_DBNR     := 100,       // Quelle: DB 100
    QUELLE_ANFANG   := 0,         // ab Datenbyte DBB 0
    QUELLE_LAENGE   := 8,         // Länge: 4 Byte
    ZIEL_DBNR       := 110,       // Ziel: DB 110
    ZIEL_ANFANG     := 2,         // ab Datenbyte DBB 6
    ZIEL_LAENGE     := 8,         // Länge: 4 Byte
    RET_VAL         := FEHLER);   // RET_VAL = Fehlercode des SFC 20 BLKMOV

END_ORGANIZATION_BLOCK
```

Bild 9-11 OB 1

Anhänge

Operanden- / Operationslisten



A.1 Operanden

Konvertierbare Operanden

Folgende Operanden werden konvertiert:

Tabelle A-1 Konvertierbare Operanden

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”A”	”Q”	”A”	”Q”
”AB”	”QB”	”AB”	”QB”
”AD”	”QD”	”AD”	”QD”
”AW”	”QW”	”AW”	”QW”
”BF”	”BN”	””	””
”D”	”D”	”DBX”	”DBX”
”DW”	”DW”	”DBW”	”DBW”
”DD”	”DD”	”DBD”	”DBD”
”DR”	”DR”	”DBB”	”DBB”
”DL”	”DL”	”DBB”	”DBB”
”E”	”I”	”E”	”I”
”EB	”IB”	”EB”	”IB”
”ED”	”ID”	”ED”	”ID”
”EW”	”IW”	”EW”	”IW”
”M”	”F”	”M”	”M”
”MB”	”FY”	”MB”	”MB”
”MD”	”FD”	”MD”	”MD”
”MW”	”FW”	”MW”	”MW”
”PW”	”PW”	”PEW/PAW”	”PIW/PQW”
”PY”	”PY”	”PEB/PAB”	”PIB/PQB”
”QB”	”OY”	”PEB/PAB”	”PIB/PQB”
”QW”	”OW”	”PEW/PAW”	”PIW/PQW”
”S”	”S”	”M”	”M”
”SD”	”SD”	”MD”	”MD”
”SW”	”SW”	”MW”	”MW”
”SY”	”SY”	”MB”	”MB”
”T”	”T”	”T”	”T”

Tabelle A-1 Konvertierbare Operanden, Fortsetzung

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”Z”	”C”	”Z”	”C”
”= <Formal-parameter>”	”= <Formal-parameter>”	”# <Formal-parameter>”	”# <Formal-parameter>”

Nicht konvertierbare Operanden

Tabelle A-2 zeigt, welche Operanden **nicht konvertiert** werden können.

Tabelle A-2 Nicht konvertierbare Operanden

S5-AWL (Deutsch)	S5-AWL (International)
”A1”	”A1”
”A2”	”A2”
”BA”	”RI”
”BB”	”RJ”
”BR”	”BR”
”BS”	”RS”
”BT”	”RT”
”CB”	”CY”
”CD”	”CD”
”CW”	”CW”
”GB”	”GY”
”GD”	”GD”
”GW”	”GW”
”SA”	”SA”

A.2 Operationen

Konvertierbare Operationen ohne Operanden

Tabelle A-3 zeigt alle S5-Operationen (ohne Operanden) in AWL, die automatisch in S7-AWL konvertiert werden:

Tabelle A-3 Konvertierbare Operationen (ohne Operanden)

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”AF”	”RA”	”CALL SFC 42”	”CALL SFC 42”
”AS”	”IA”	”CALL SFC 41”	”CALL SFC 41”
”BEA”	”BEU”	”BEA”	”BEU”
”BEB”	”BEC”	”BEB”	”BEC”
”+D”	”+D”	”+D”	”+D”
”–D”	”–D”	”–D”	”–D”
”!=D”	”!=D”	”==D”	”==D”
”><D”	”><D”	”<>D”	”<>D”
”>D”	”>D”	”>D”	”>D”
”>=D”	”>=D”	”>=D”	”>=D”
”<D”	”<D”	”<D”	”<D”
”<=D”	”<=D”	”<=D”	”<=D”
”DED”	”DED”	”BTD”	”BTD”
”DEF”	”DEF”	”BTI”	”BTI”
”DUD”	”DUD”	”DTB”	”DTB”
”DUF”	”DUF”	”ITB”	”ITB”
”ENT”	”ENT”	”ENT”	”ENT”
”+F”	”+F”	”+I”	”+I”
”–F”	”–F”	”–I”	”–I”
”:F”	”:F”	”/I”	”/I”
”xF”	”xF”	”*I”	”*I”
”!=F”	”!=F”	”==I”	”==I”
”><F”	”><F”	”<>I”	”<>I”
”>F”	”>F”	”>I”	”>I”
”>=F”	”>=F”	”>=I”	”>=I”
”<F”	”<F”	”<I”	”<I”
”<=F”	”<=F”	”<=I”	”<=I”
”FDG”	”FDG”	”DTR”	”DTR”
”+G”	”+G”	”+R”	”+R”
”–G”	”–G”	”–R”	”–R”
”:G”	”:G”	”/R”	”/R”
”xG”	”xG”	”*R”	”*R”
”!=G”	”!=G”	”==R”	”==R”
”><G”	”><G”	”<>R”	”<>R”
”>G”	”>G”	”>R”	”>R”

Tabelle A-3 Konvertierbare Operationen (ohne Operanden), Fortsetzung

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”>=G”	”>=G”	”>=R”	”>=R”
”<G”	”<G”	”<R”	”<R”
”<=G”	”<=G”	”<=R”	”<=R”
”GFD”	”GFD”	”RND”	”RND”
”KEW”	”CFW”	”INVI”	”INVI”
”KZD”	”CSD”	”NEGD”	”NEGD”
”KZW”	”CSW”	”NEGI”	”NEGI”
”O”	”O”	”O”	”O”
”O(”	”O(”	”O(”	”O(”
”OW”	”OW”	”OW”	”OW”
”STP”	”STP”	”CALL SFC 46”	”CALL SFC 46”
”STS”	”STS”	”CALL SFC 46”	”CALL SFC 46”
”STW”	”STW”	”CALL SFC 46”	”CALL SFC 46”
”TAK”	”TAK”	”TAK”	”TAK”
”U(”	”A(”	”U(”	”A(”
”UW”	”AW”	”UW”	”AW”
”XOW”	”XOW”	”XOW”	”XOW”
”)”	”)”	”)”	”)”
”***”	”***”	”NETWORK”	”NETWORK”

Konvertierbare Operationen mit Operanden

Tabelle A-4 zeigt alle S5-Operationen (mit Operanden) in AWL, die automatisch in S7-AWL konvertiert werden:

Tabelle A-4 Konvertierbare Operationen (mit Operanden)

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”A”	”C”	”AUF”	”OPN”
”ADD BF” ”ADD DH” ”ADD KF”	”ADD BF” ”ADD DH” ”ADD KF”	”+” ”+” ”+”	”+” ”+” ”+”
”AX”	”CX”	”AUF”	”OPN”
”B”	”DO”	”Anweisungsfolge für indirekte Adressierung”	”Anweisungsfolge für indirekte Adressierung”
”BA”	”BA”	””	””
”BAB”	”DOC”	”SPB”	”JC”
”D”	”D”	”DEC”	”DEC”
”E”	”G”	”CALL SFC 22”	”CALL SFC 22”
”EX”	”GX”	”CALL SFC 22”	”CALL SFC 22”
”FR”	”FR”	”FR”	”FR”
”I”	”I”	”INC”	”INC”
”L”	”L”	”L”	”L”

Tabelle A-4 Konvertierbare Operationen (mit Operanden), Fortsetzung

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”LC”	”LD”	”LC”	”LC”
”NOP”	”NOP”	”NOP”	”NOP”
”O”	”O”	”O”	”O”
”ON”	”ON”	”ON”	”ON”
”P”	”TB”	”SET; U”	”SET; A”
”PN”	”TBN”	”SET; UN”	”SET; AN”
”R”	”R”	”R”	”R”
”RB”	”RB”	”R”	”R”
”RD”	”RD”	”R”	”R”
”RLD”	”RLD”	”RLD”	”RLD”
”RLW”	”RLW”	”RLW”	”RLW”
”RRD”	”RRD”	”RRD”	”RRD”
”RRW”	”RRW”	”RRW”	”RRW”
”RU”	”RU”	”SET; R”	”SET; R”
”S”	”S”	”S”	”S”
”SA”	”SF”	”SA”	”SF”
”SAR”	”SFD”	”SA” Timer ”ZR” Zähler	”SF” Timer ”CD” Counter
”SE”	”SD”	”SE”	”SD”
”SI”	”SP”	”SI”	”SP”
”SLD”	”SLD”	”SLD”	”SLD”
”SLW”	”SLW”	”SLW”	”SLW”
”SPA”	”JU”	”SPA”	”JU”
”SPB”	”JC	”SPB”	”JC”
”SPM”	”JM”	”SPM”	”JM”
”SPN”	”JN”	”SPN”	”JCN”
”SPO”	”JO”	”SPO”	”JO”
”SPP”	”JP”	”SPP”	”JP”
”SPR”	”JUR”	”SPA”	”JU”
”SPS”	”JOS”	”SPS”	”JOS”
”SPZ”	”JZ”	”SPZ”	”JZ”
”SRD”	”SRD”	”SRD”	”SRD”
”SRW”	”SRW”	”SRW”	”SRW”
”SS”	”SS”	”SS”	”SS”
”SSV”	”SSU”	”SS” Timer ”ZV” Zähler	”SS” Timer ”CU” Counter
”SU”	”SU”	”SET; S”	”SET; S”
”SV”	”SE”	”SV”	”SE”
”SVD”	”SSD”	”SSD”	”SSD”
”SVW”	”SSW”	”SSI”	”SSI”
”SVZ”	”SEC”	”SV” Timer ”S” Zähler	”SE” Timer ”S” Counter

Tabelle A-4 Konvertierbare Operationen (mit Operanden), Fortsetzung

S5-AWL (Deutsch)	S5-AWL (International)	S7-AWL (Deutsch)	S7-AWL (International)
”T”	”T”	”T”	”T”
”TNB”	”TNB”	”CALL SFC 20”	”CALL SFC 20”
”TNW”	”TNW”	”CALL SFC 20”	”CALL SFC 20”
”U”	”A”	”U”	”A”
”UN”	”AN”	”UN”	”AN”
”ZR”	”CD”	”ZR”	”CD”
”ZV”	”CU”	”ZV”	”CU”
”=”	”=”	”=”	”=”

Nicht konvertierbare Operationen

Die folgende Tabelle zeigt die S5-AWL-Operationen, die nicht automatisch konvertiert werden.

Tabelle A-5 Nicht konvertierbare Operationen

S5-AWL (Deutsch)	S5-AWL (International)
”AAS”	”IAI”
”AAF”	”RAI”
”ABR”	”ABR”
”ACR”	”ACR”
”AFF”	”RAE”
”AFS”	”IAE”
”ASM”	”ASM”
”BAF”	”BAF”
”BAS”	”BAS”
”BI” (nur konvertierbar für Parameterart D/Konstante)	”DI” (nur konvertierbar für Parameterart D/Konstante)
”BLD”	”BLD”
”LB”	”LB”
”LD”	”LD”
”LD=<Formalparameter>” (nur konvertierbar für Parameterart D/Konstante)	”LD=<Formalparameter>” (nur konvertierbar für Parameterart D/Konstante)
”LDI”	”LDI”
”LIM”	”LIM”
”LIR”	”LIR”
”LRB”	”LRB”
”LRD”	”LRD”
”LRW”	”LRW”
”LW”	”LW”
”LW=<Formalparameter>” (nur konvertierbar für Parameterart D/Konstante)	”LW=<Formalparameter>” (nur konvertierbar für Parameterart D/Konstante)
”MA1”	”MA1”
”MAB”	”MAB”

Tabelle A-5 Nicht konvertierbare Operationen, Fortsetzung

S5-AWL (Deutsch)	S5-AWL (International)
”MAS”	”MAS”
”MBA”	”MBA”
”MBR”	”MBR”
”MBS”	”MBS”
”MSA”	”MSA”
”MSB”	”MSB”
”SEF”	”SEE”
”SES”	”SED”
”SIM”	”SIM”
”TB”	”TB”
”TDI”	”TDI”
”TIR”	”TIR”
”TSC”	”TSC”
”TSG”	”TSG”
”TRB”	”TRB”
”TRD”	”TRD”
”TRW”	”TRW”
”TW”	”TW”
”TXB”	”TXB”
”TXW”	”TXW”
”UBE”	”UBE”

Literaturverzeichnis

B

/21/ Broschüre: *Automatisierungssystem S7/M7*, Dezentralisieren mit PROFIBUS-DB und AS-I

/30/ Fibel: *Automatisierungssystem S7-300,* Einfach aufbauen und programmieren

/70/ Handbuch: *Automatisierungssystem S7-300,* Aufbauen, CPU-Daten

/71/ Referenzhandbuch: *Automatisierungssysteme S7-300, M7-300* Baugruppendaten

/72/ Operationsliste: *Automatisierungssystem S7-300*

/100/ Installationshandbuch: *Automatisierungssysteme S7-400, M7-400,* Aufbauen

/101/ Referenzhandbuch: *Automatisierungssysteme S7-400, M7-400* Baugruppendaten

/102/ Operationsliste: *Automatisierungssystem S7-400*

/231/ Benutzerhandbuch: *Basissoftware für S7 und M7, STEP 7*

/232/ Handbuch: *AWL für S7-300/400, Bausteine programmieren*

/233/ Handbuch: *KOP für S7-300/400,* Bausteine programmieren

/234/ Programmierhandbuch: *Systemsoftware für S7-300/400* Programmentwurf

/235/ Referenzhandbuch: *Systemsoftware für S7-300/400* System- und Standardfunktionen

/236/ Handbuch: *FUP für S7-300/400,* Bausteine programmieren

/249/ Handbuch: *CFC für S7 / M7, Band 2*

/250/ Handbuch: *SCL für S7-300/400,* Bausteine programmieren

/251/ Handbuch: *GRAPH für S7-300/400,* Ablaufsteuerungen programmieren

/252/ Handbuch: *HiGraph für S7-300/400,* Zustandsgraphen programmieren

/253/ Handbuch: *C für S7-300/400,* C-Programme erstellen

/254/ Handbuch: *CFC für S7 und M7,* Band 1

/270/ Handbuch: *S7-PDIAG für S7-300/400*
Prozeßdiagnose für KOP, FUP und AWL projektieren

/271/ Handbuch: *NETPRO*
Netze grafisch projektieren

/280/ Programmierhandbuch: *Systemsoftware für M7-300/400,*
Programmentwurf

/281/ Referenzhandbuch: *Systemsoftware für M7-300/400,*
System- und Standardfunktionen

/282/ Benutzerhandbuch: *Systemsoftware für M7-300/400,*
Installieren und Bedienen

/290/ Benutzerhandbuch: *ProC/C++ für M7-300/400,*
C-Programme erstellen

/291/ Benutzerhandbuch: *ProC/C++ für M7-300/400,*
Debugger für C-Programme

/500/ Handbuch: *SIMATIC NET,*
NCM S7 für Industrial Ethernet

/501/ Handbuch: *SIMATIC NET*
NCM S7 für PROFIBUS

/800/ *DOCPRO*
Schaltbücher normgerecht erstellen (nur auf CD)

/801/ *TeleService für S7, C7 und M7*
Fernwartung eines Automatisierungssystems (nur auf CD)

/802/ *S7-PLCSIM*
Programmtest mit simulierter S7-CPU (nur auf CD)

/803/ Referenzhandbuch: *Systemsoftware für S7-300/400*
STEP 7 Standardfunktionen Teil 2 (nur auf CD)

Glossar

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden oder Operandenbereich. Beispiele: Eingang E12.1; Merkerwort MW 25; Datenbaustein DB 3.

Aktualparameter

Aktualparameter ersetzen beim Aufruf eines Funktionsbausteins (FB) oder einer Funktion (FC) die Formalparameter, z.B. wird der Formalparameter ”START” ersetzt durch den Aktualparameter ”E 3.6”.

Anweisung

Eine Anweisung ist die kleinste selbständige Einheit eines in einer textuellen Sprache erstellten Anwenderprogramms. Sie stellt eine Arbeitsvorschrift für den Prozessor dar.

Anweisungsliste (AWL)

Die Anweisungsliste ist eine maschinennahe textuelle Programmiersprache.

B

Baustein

Bausteine sind durch ihre Funktion, ihre Struktur oder ihren Verwendungszweck abgegrenzte Teile des Anwenderprogrammes. Es gibt bei STEP 7:

- Codebausteine (FB, FC, OB, SFB, SFC),
- Datenbausteine (DB, SDB) und
- anwenderdefinierte Datentypen (UDT).

Bausteinaufruf

Als Bausteinaufruf bezeichnet man die Verzweigung der Programmbearbeitung in den aufgerufenen Baustein.

Bausteinparameter

Bausteinparameter sind Platzhalter innerhalb mehrfach nutzbarer Bausteine, die beim Aufruf des betreffenden Bausteins mit aktuellen Werten versorgt werden.

C

Codebaustein

Ein Codebaustein ist bei SIMATIC S7 ein Baustein, der einen Teil des STEP 7-Anwenderprogramms enthält.

Im Gegensatz dazu enthält ein Datenbaustein nur Daten. Es gibt folgende Codebausteine: Organisationsbausteine (OB), Funktionsbausteine (FB), Funktionen (FC), Systemfunktionsbausteine (SFB), Systemfunktionen (SFC).

Compiler

Als Compiler bezeichnet man ein Übersetzungsprogramm zur Übersetzung eines in einer höheren Programmiersprache geschriebenen Programmes in den Maschinencode, mit dem die CPU arbeitet.

D

Daten, statisch

Statische Daten sind Lokaldaten eines Funktionsbausteins, die im Instanzdatenbaustein gespeichert werden und deshalb bis zur nächsten Bearbeitung des Funktionsbausteins erhalten bleiben.

Daten, temporär

Temporäre Daten sind Lokaldaten eines Bausteins, die während der Bearbeitung des Bausteins im L-Stack abgelegt werden und nach der Bearbeitung des Bausteins nicht mehr verfügbar sind.

Datenbaustein (DB)

Datenbausteine sind Datenbereiche im Anwenderprogramm, die Anwenderdaten enthalten. Es gibt globale Datenbausteine, auf die von allen Codebausteinen zugegriffen werden kann und es gibt Instanz-Datenbausteine, die einem bestimmten FB-Aufruf zugeordnet sind. Datenbausteine enthalten im Gegensatz zu allen anderen Bausteinen keine Anweisungen.

Datentyp

Mit Hilfe eines Datentyps können Sie festlegen, wie der Wert einer Variablen oder Konstanten im Anwenderprogramm verwendet werden soll. Dem Anwender stehen bei SIMATC S7 zwei Arten von Datentypen nach IEC 1131-3 zur Verfügung: elementare Datentypen und zusammengesetzte Datentypen.

Datentyp, elementar

Elementare Datentypen sind vordefinierte Datentypen gemäß IEC 1131-3, z.B. Datentyp BOOL definiert eine binäre Variable (”Bit”), Datentyp INT definiert eine 16-Bit-Festpunktzahl-Variable.

Datentyp, zusammengesetzt

Zusammengesetzte Datentypen werden vom Anwender mit der Datentypdeklaration geschaffen. Sie haben keinen eigenen Namen und sind deshalb nicht mehrfach verwendbar. Man unterscheidet zwischen Feldern und Strukturen, auch die Datentypen *String* und *Date and Time* zählen dazu.

Deklarationsteil

Im Deklarationsteil werden die Lokaldaten eines Codebausteins deklariert, wenn die Programmerstellung mit einem Texteditor erfolgt.

F

Formalparameter

Ein Formalparameter ist ein Platzhalter für den ”tatsächlichen” Parameter (Aktualparameter) bei parametrierbaren Codebausteinen. Bei Funktionsbausteinen und Funktionen werden die Formalparameter vom Anwender deklariert, bei System-Funktionsbausteinen und System-Funktionen sind sie bereits vorhanden.

Beim Aufruf des Bausteins wird dem Formalparameter ein Aktualparameter zugeordnet, so daß der aufgerufene Baustein mit dessen aktuellem Wert arbeitet. Die Formalparameter zählen zu den Lokaldaten des Bausteins und unterteilen sich in Eingangs-, Ausgangs- und Durchgangsparameter.

Funktion (FC)

Eine Funktion (FC) ist gemäß IEC 1131-3 ein Codebaustein ohne Gedächtnis. Eine Funktion bietet die Möglichkeit der Übergabe von Parametern im Anwenderprogramm. Dadurch eignen sich Funktionen zur Programmierung von häufig wiederkehrenden komplexen Funktionen, z. B. Berechnungen. Da kein Gedächtnis vorhanden ist, müssen die berechneten Werte direkt nach dem FC-Aufruf weiterverarbeitet werden.

Funktionsbaustein (FB)

Ein Funktionsbaustein ist gemäß IEC 1131-3 ein Codebaustein mit statischen Daten. Ein Funktionsbaustein bietet die Möglichkeit der Übergabe von Parametern im Anwenderprogramm. Dadurch eignen sich Funktionsbausteine zur Programmierung von häufig wiederkehrenden komplexen Funktionen, z. B. Regelungen, Betriebsartenanwahl. Da ein FB über ein Gedächtnis (Instanz-Datenbaustein) verfügt, kann auf dessen Parameter (z. B. Ausgänge) zu jeder Zeit an jeder beliebigen Stelle im Anwenderprogramm zugegriffen werden.

G

Globaldaten

Globaldaten sind Daten, die von jedem Codebaustein (FC, FB, OB) aus ansprechbar sind. Im einzelnen sind das Merker M, Eingänge E, Ausgänge A, Zeiten, Zähler und Elemente von Datenbausteinen DB. Auf Globaldaten kann entweder absolut oder symbolisch zugegriffen werden.

I

Instanz

Mit ”Instanz” wird der Aufruf eines Funktionsbaustein bezeichnet; dabei ist ihm ein Instanz-Datenbaustein zugeordnet.

Instanz-Datenbaustein

Ein Instanz-Datenbaustein speichert die Formalparameter und statischen Daten von Funktionsbausteinen. Ein Instanz-Datenbaustein kann einem FB-Aufruf oder einer Aufrufhierarchie von Funktionsbausteinen zugeordnet sein.

K

Konfigurieren

Auswählen und Zusammenstellen einzelner Komponenten eines Automatisierungssystems bzw. Installieren von benötigter Software und Anpassen an den speziellen Einsatz (z.B. durch Parametrieren der Baugruppen).

L

Lokaldaten

Lokaldaten sind die einem Codebaustein zugeordneten Daten, die in seinem Deklarationsteil bzw. seiner Variablendeklaration deklariert werden. Sie umfassen (bausteinabhängig): Formalparameter, statische Daten, temporäre Daten.

M

Makro

Ein Makro ist eine Folge von Befehlen, die ablaufoptimiert zu einem mnemotechnischen Aufruf zusammengefaßt sind.

O

Online-Hilfe

STEP 7 bietet Ihnen die Möglichkeit, sich während des Arbeitens mit der Programmiersoftware kontextabhängige Hilfetexte am Bildschirm anzeigen zu lassen.

Operand

Ein Operand ist Teil einer STEP 7-Anweisung und sagt aus, womit der Prozessor etwas tun soll. Er kann sowohl absolut als auch symbolisch adressiert werden.

Operation

Eine Operation ist Teil einer STEP 7-Anweisung und sagt aus, was der Prozessor tun soll.

Organisationsbaustein (OB)

Organisationsbausteine bilden die Schnittstelle zwischen dem Betriebssystem der CPU und dem Anwenderprogramm. In den Organisationsbausteinen wird die Reihenfolge der Bearbeitung des Anwenderprogrammes festgelegt.

P

Parametrieren

Unter Parametrieren versteht man das Einstellen des Verhaltens einer Baugruppe.

Peripherie, dezentral

Dezentrale Peripherie sind vom zentralen Baugruppenträger räumlich abgesetzte Analog- und Digitalbaugruppen. Charakteristisch für die dezentrale Peripherie ist die Aufbautechnik. Zielsetzung dieser Aufbautechnik ist das Einsparen von Verdrahtungsaufwand (und damit Kosten) durch prozeßnahes Einsetzen der Peripheriebaugruppen.

Programmiersprache

Eine Programmiersprache dient zur Erstellung von Anwenderprogrammen und stellt dazu einen bestimmten Sprachvorrat in Form von grafischen oder textuellen Anweisungen zur Verfügung. Diese Anweisungen werden vom Anwender mit einem Editor eingegeben und in ein lauffähiges Anwenderprogramm übersetzt.

Projekt

Ein Projekt ist ein Behälter für alle Objekte einer Automatisierungslösung, unabhängig von der Anzahl der Stationen, Baugruppen und deren Vernetzung.

R

Remanenz

Daten werden remanent genannt, wenn sie nach dem Ausfall der Versorgungsspannung den gleichen Wert haben wie vor dem Spannungsausfall. Die Remanenz wird erreicht durch zwei Pufferungen: Spannungspufferung und Backup-Pufferung.

S

S7-Programm

Ein Behälter für Bausteine, Quellen und Pläne für programmierbare S7-Baugruppen, der auch die Symboltabelle enthält.

Symbol

Ein Symbol ist ein vom Anwender unter Berücksichtigung bestimmter Syntaxvorschriften definierter Name. Dieser Name kann nach der Festlegung, wofür er stehen soll (z.B. Variable, Datentyp, Sprungmarke, Baustein) bei der Programmierung und beim Bedienen und Beobachten verwendet werden.

Beispiel: Operand E 5.0, Datentyp BOOL, Symbol Taster NOTAUS.

Symbolik

Es wird unterschieden zwischen globaler Symbolik und bausteinlokaler Symbolik. Global vereinbarte Symbole sind in allen Programmteilen bekannt, das vergebene Symbol muß für das gesamte Anwenderprogramm eindeutig sein. Bausteinlokale Symbole sind nur innerhalb des Bausteins bekannt, in dem sie vereinbart werden.

Symboltabelle

Tabelle zur Zuordnung von Symbolen zu Adressen für Globaldaten und Bausteinen. Beispiele: NOTAUS (Symbol), E1.7 (Adresse) oder Regler (Symbol), SFB 24 (Baustein)

V

Variable

Eine Variable definiert ein Datum mit variablem Inhalt, das im STEP 7-Anwenderprogramm verwendet werden kann. Eine Variable besteht aus einem Operanden und einem Datentyp und kann mit einem Symbol gekennzeichnet werden.

Z

Zeiger

Ein Zeiger ist eine Variable, die keinen bestimmten Wert sondern die Adresse einer anderen Variablen enthält. Bei Zeigeroperationen ist es erforderlich, daß der Typ auf der rechten Seite des Operators mit dem Typ auf der linken Seite übereinstimmt.

Stichwortverzeichnis

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q